AD-A046 348   ARGONNE NATIONAL LAB ILL   F/G 13/2

AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)

APR 77 D J BINGAMAN

UNCLASSIFIED   CEEDO-TR-76-34   NL

1 OF 3

AD
A046348

CEEDO-TR-76-34

12

B.S.

AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS—SHORT-TERM EMISSION/DISPERSION COMPUTER CODE DOCUMENTATION

ARGONNE NATIONAL LABORATORY
9700 SOUTH CASS AVENUE
ARGONNE, ILLINOIS 60439

APRIL 1977

CEEDO

FINAL REPORT FOR PERIOD
1 JULY 1975-1 JANUARY 1977

Approved for public release; distribution unlimited

CIVIL AND ENVIRONMENTAL
ENGINEERING DEVELOPMENT OFFICE
(AIR FORCE SYSTEMS COMMAND)
TYNDALL AIR FORCE BASE
FLORIDA 32403

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CEEDO-TR-76-34
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-TERM EMISSION/DISPERSION COMPUTER CODE DOCUMENTATION.
5. TYPE OF REPORT & PERIOD COVERED: Final Report, 1 July 1975 to 1 Jan 1977
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Dorothy J. Bingaman
8. CONTRACT OR GRANT NUMBER(s): CEEDO ~~Project Order~~ 76-0003
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Argonne National Laboratory, 9700 South Cass Avenue, Argonne IL 60439

   per ogl 1498 verification
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62601F, 1900, 5A03
11. CONTROLLING OFFICE NAME AND ADDRESS: Civil Environmental & Engineering Development Office, Det 1 HQ ADTC, Tyndall AFB FL 32403
12. REPORT DATE: April 1977
13. NUMBER OF PAGES: 209
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 206p.
15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

Ed Mr Smith

DDC RECEIVED NOV 14 1977 B

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

9/312 739 7711 X ~~3026~~, 3029

Baird Capt Bar 8/970 2996

18. SUPPLEMENTARY NOTES

Available in DDC.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Aircraft Assessment
Airport Models
Air Pollution
Dispersion Model
Computer Code

872 4141

Mary Davis 872 - ~~2206~~ 4013

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Air Force contracted with Argonne National Laboratory to develop a series of computer programs called the Air Quality Assessment Model (AQAM). The source emissions inventory routine of AQAM was designed to handle complex emission sources with emphasis on aircraft. A short term emission/dispersion model for hourly air quality predictions and a long term emission/dispersion model for monthly and annual predictions are also in AQAM.

This report documents only the short term model. Flow charts, computer

over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

033550 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Block 20 continued

listings, and brief descriptions of each subroutine are included. They are intended for readers with a computer background who wish to examine or alter the computer code.

ACCESSION for		
NTIS	White Section	☑
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This report documents work performed during the period 1 July 1975 through December 1976 by Argonne National Laboratory. The technical work for this effort was performed under the auspices of the Air Force Civil Engineering Center (AFSC) which on 8 April 1977, reorganized into Detachment 1 (CEEDO) HQ ADTC, Tyndall Air Force Base, Florida, 32403. Captain Dennis F. Naugle, CEEDO/ECA, managed the program.

This report has been reviewed by the Information Officer and is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Dennis F. Naugle
DENNIS F. NAUGLE, Capt, USAF, BSC
Chief, Environmental Modeling Branch

Peter S. Daley
PETER S. DALEY, Maj, USAF, BSC
Chief, Environmental Assessment Research Division

JOSEPH S. PIZZUTO, Col, USAF, BSC
Commander

Peter A. Crowley
PETER A. CROWLEY, Maj, USAF, BSC
Director of Environics

(The reverse of this page is blank)

TABLE OF CONTENTS

TABLE OF CONTENTS (CONCLUDED)

SECTION I
INTRODUCTION

Argonne National Laboratory (ANL) has developed an Air Quality Assessment Model (AQAM) for airbase operations under contract to the U.S. Air Force Civil Engineering Center (AFCEC) designed to simulate the emission of pollutants from sources on an airbase and the dispersion of these emissions in the atmosphere so as to enable calculation of pollutant concentrations over a grid of ground level receptors. These models are comprised of four physically separate computer codes, of which three must be operated by the user. The fourth code prepares a magnetic tape containing long term stability-time-wind roses for use by the long term climatological type air pollution model. This code is operated on request by the USAF Environmental Technical Applications Center in Washington, D.C. and the resultant magnetic tapes containing the climatological information is shipped to the user. The other three codes, developed by ANL, consist of the

- Source Inventory Model (SRCINV)
- Short Term Emission/Dispersion Model
- Long Term Emission/Dispersion Model

This report constitutes the computer code documentation for the second of these - the Short Term Emission/Dispersion Model. A separate computer code documentation manual (Reference 1) is available for SRCINV. Documentation for the Long Term Emission/Dispersion Model is currently being prepared and will be available shortly. A companion document to these reports - Operator's Guide (Reference 2) of the Air Quality Assessment Model for airbase operations - consists of a detailed discussion of the various functional parts of the computer programs and the input/output requirements. A second companion report (Reference 3) discusses the technical and theoretical basis underlying AQAM and presents and describes equations and algorithms used in the various AQAM sub-models.

The intended purpose of the present document is to provide a computer programmer with sufficient information so that he can study the code and make changes or modifications to it where required.

Table 1 contains a list of all routines contained in the Short Term Model in alphabetical order together with a brief description. More detailed descriptions of each routine, together with flow charts and computer code listings with comments that are intended to link listings to flow charts, are given on subsequent pages. It is hoped that this information, when combined with that given in References 1, 2, and 3, will enable a programmer to understand and make changes to the code when desired.

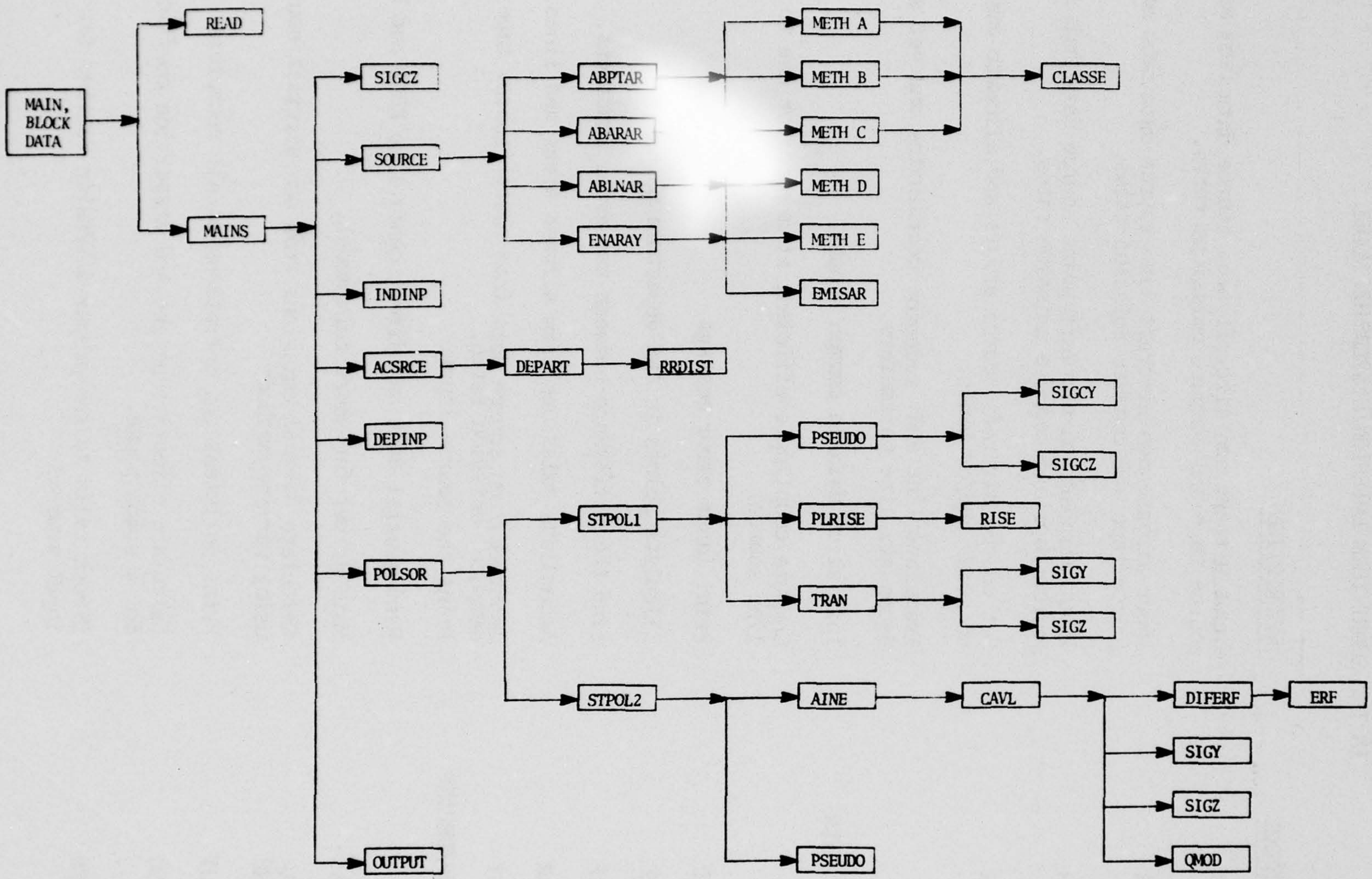


Figure 1. Schematic Flow Diagram of Short Term Model

TABLE 1. LIST OF ALL PROGRAMS AND SUBPROGRAMS IN THE SHORT TERM EMISSION/DISPERSION MODEL

SUBROUTINE	DESCRIPTION
ABARAR	Input airbase non-aircraft area source data from master source tape and compute emissions rates.
ABLNAR	Input airbase non-aircraft line source data from master source tape and compute emission rates.
ABPTAR	Input airbase non-aircraft point source data from master source tape and compute emission rates.
ACSRCE	Set up the aircraft source arrays and allocate emissions to areas and/or lines.
AINE	Translate line and receptor coordinates and set all necessary line parameters.
BLOCK DATA	Initialize data in common blocks.
CAVL	Compute coupling coefficient at a receptor due to a line source.
CLASSE	Print input error message.
DEPART	Calculate points in the departure path.
DIFERF	Find the difference between two error functions.
EMISAR	Accumulate emissions from airbase areas and lines.
ENARAY	Input environ source data from master source tape and compute emission rates.
INDINP/DEPINP	Print the source input.
MAIN	Read general data and direct control to READ and MAINS.
MAINS	Main driver for short term model.
METHA-METHE	Calculate diurnal emissions from non-aircraft sources using varying methods.
OUTPUT	Print pollutant concentrations at all receptors.
PLRISE	Calculate effective height and dispersion coefficients for a stack plume.
POLSOR	Direct calls to the proper diffusion routine for all input sources.

TABLE 1. LIST OF ALL PROGRAMS AND SUBPROGRAMS IN THE SHORT TERM EMISSION/DISPERSION MODEL (CONCLUDED)

PSEUDO	Call functions to find virtual distance from source to pseudo upwind point.
QMOD	Compute linear distribution of pollution along a runway.
READ	Read master source tape.
RISE	Calculate plume rise.
RRDIST	Calculate length of runway necessary for takeoff.
SIGY/SIGCY	Calculate horizontal dispersion or corresponding virtual distance.
SIGZ/SIGCZ	Calculate vertical dispersion or corresponding virtual distance.
SOURCE	Driver for non-aircraft emission routines.
STPOL1	Determine pollutant concentrations from point and area sources.
STPOL2	Determine pollutant concentrations from line sources.
TRAN	Calculate the coupling coefficient at a receptor due to a point or area source.

SUBROUTINE ABARAR

Purpose:

1. To read from the master source type all data needed to define airbase non-aircraft area sources.
2. To compute the emission rates due to evaporative hydrocarbons, space heating, off-road vehicles, and military and civilian vehicles.

Input:

If the diurnal distribution cards are input, an additional parameter, IOPT, is read here to choose the method of distribution of those evaporative hydrocarbons not using the default of a uniform distribution.

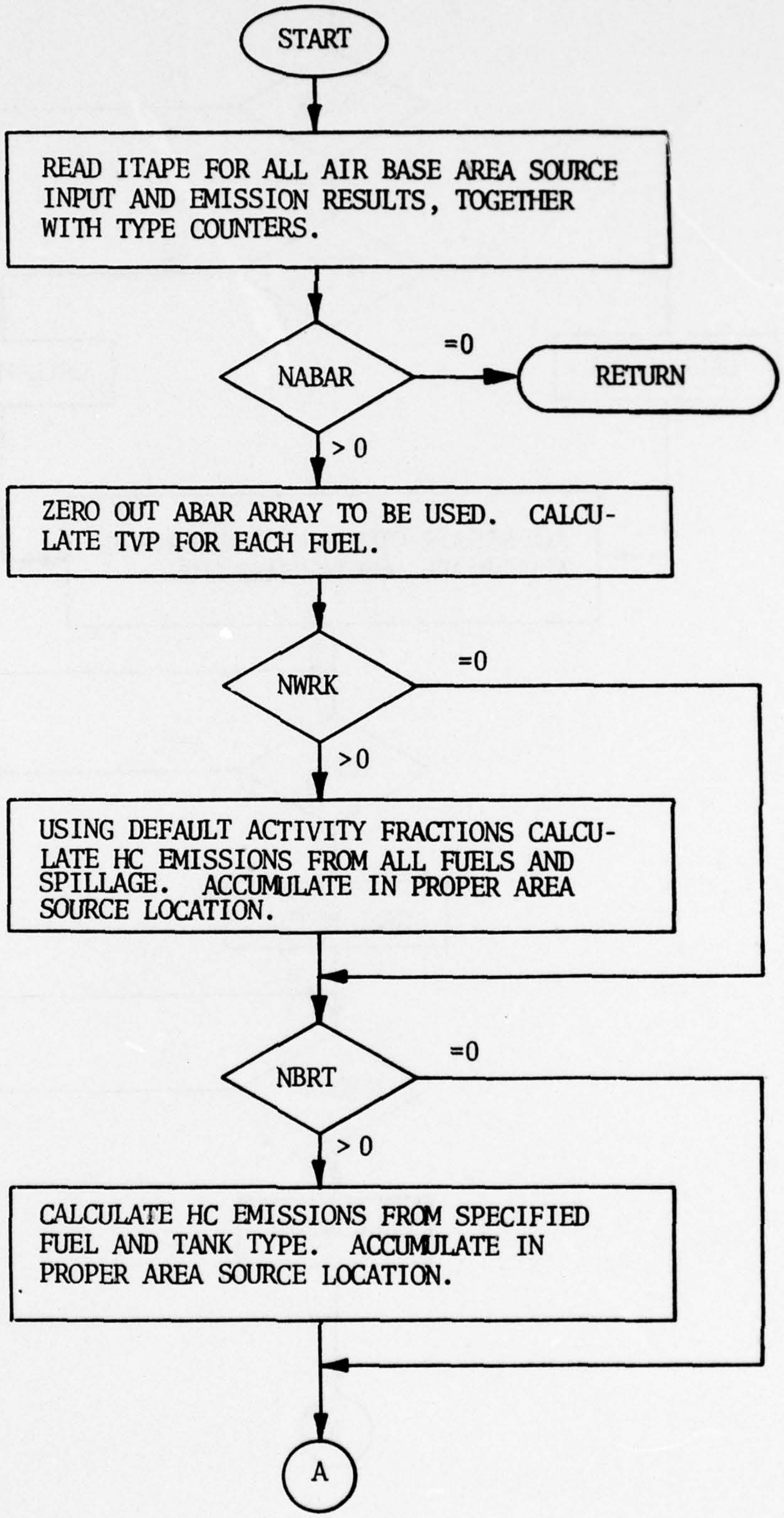
Output:

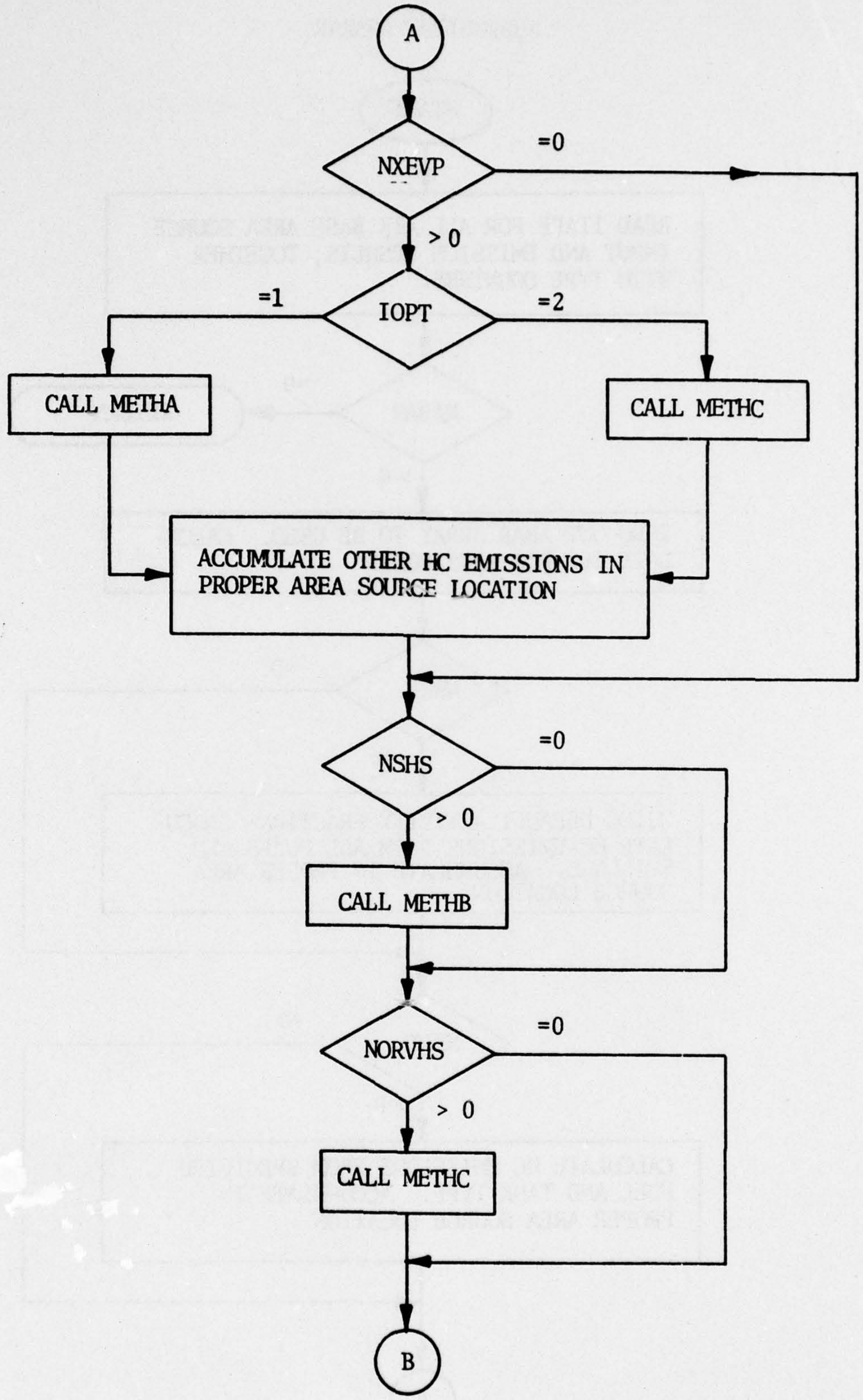
The array, ABAR, is filled with geometry and emission data for airbase non-aircraft area sources.

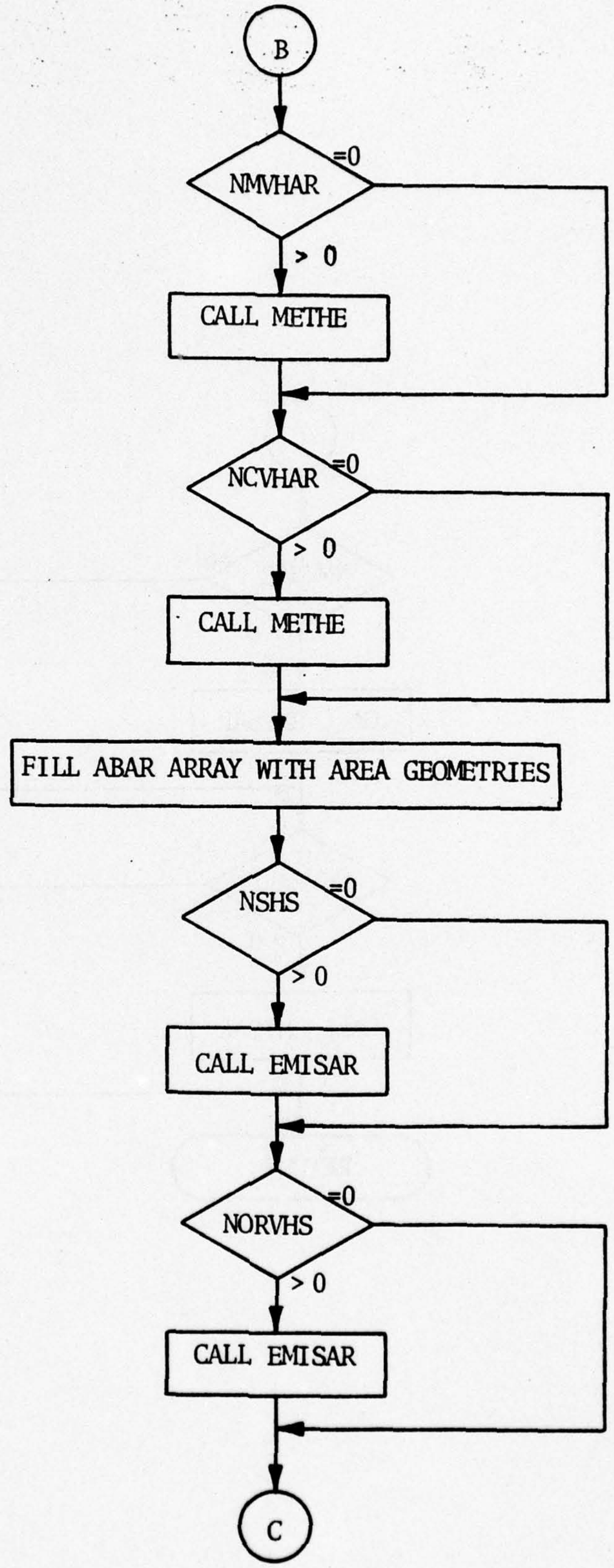
Subroutines Called:

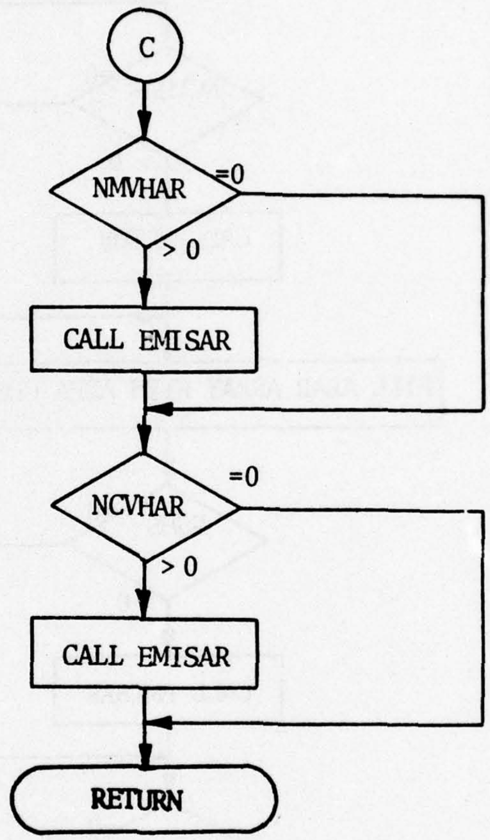
METHA, METHB, METHC, METHE, EMISAR

SUBROUTINE ABARAR




A
NXEVP
=0
> 0
IOPT
=1
=2
CALL METHA
CALL METHC
ACCUMULATE OTHER HC EMISSIONS IN PROPER AREA SOURCE LOCATION
NSHS
=0
> 0
CALL METHB
NORVHS
=0
> 0
CALL METHC
B


B
NMVHAR
=0
> 0
CALL METHE
NCVHAR
=0
> 0
CALL METHE
FILL ABAR ARRAY WITH AREA GEOMETRIES
NSHS
=0
> 0
CALL EMISAR
NORVHS
=0
> 0
CALL EMISAR
C


C
NMVHAR
=0
> 0
CALL EMISAR
NCVHAR
=0
> 0
CALL EMISAR
RETURN

```
      SUBROUTINE ABARAR
C
C     TIHS ROUTINE COMPUTES THE EMISSION RATES FOR
C     ALL AIRBASE AREAS
C         NWRK   = NO. OF HYDROCARBON WORKING LOSSES
C         NBRT   = NO. OF HYDROCARBON BREATHING LOSSES
C         NXEVP  = NO. OF OTHER EVAPORATIVE HYDROCARBON SOURCES
C         NSHS   = NO. OF SPACE HEATING SOURCES
C         NCRVHS = NO. OF OFF-ROAD VEHICLE SOURCES
C         NMVHAR = NO. OF MILITARY VEHICLE AREA SOURCES
C         NCVHAR = NO. OF CIVILIAN VEHICLE AREA SOURCES
C
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON/MONMET/TMBAR,WSMBAR,AMDMBR,DTMBAR
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                 ABPT(16,150),ABAR(11,100),ABLN(14,100)
      DIMENSION ABARGM(7,100),HCWRK(10,50),HCBRT(5,100),HCEVP(3,50),
     . FLHCUR(7),TVP(7)
      EQUIVALENCE (SORGM(1),ABARGM(1)),(SORGM( 701),HCWRK(1)),
     .      (SORGM(1201),HCBRT(1)), (SORGM(1701),HCEVP(1))
      LOC1=2
      LOC2=2
      NGEOM=0
      IPT=0
      NSRCE=0
      I1=17
      I2=300
C
      READ(ITAPE) NABAR,NTOT,NWRK,NBRT,NXEVP,NSHS,NORVHS,
     .  NMVHAR,NCVHAR,NABARS,((ABARGM(I,N),I=1,7),N=1,NABAR),
     . ((HCWRK(I,N),I=1,10),N=1,NWRK),
     .  ((HCBRT(I,N),I=1,5),N=1,NBRT),
     .  ((HCEVP(I,N),I=1,3),N=1,NXEVP),
     .  ((SORCE(I,N),I=1,NTOT),N=1,NABARS)
C
      IF (NABAR.EQ.0) GO TO 1100
C
      NHI=IHR2
      IF(IHR1.GT.IHR2) NHI=24+IHR2
      HRS=NHI-IHR1+1
      DO 10 N=1,NABAR
      DO 10 I=1,NPLTS
      ABAR(I+5,N)=0.0
   10 CONTINUE
      T=5./9.*(TMBAR-32.0)+273.
      DO 20 J=1,7
      TVP(J)=EXP(ALPHA(J)-BETA(J)/T)
   20 CONTINUE
C
      IF (NWRK.EQ.0) GO TO 100
C     USING DEFAULT ACTIVITY FRACTIONS CALCULATE HC
C     EMISSIONS FROM ALL FUELS AND SPILLAGE.
C     ACCUMULATE IN ABAR ARRAY
C
      DO 50 N=1,NWRK
```

```
      HC=0.
      FRC=0.
      DO 40 J=1,7
      FLHOUR(J)=0.
      DO 30 I=IHR1,NHI
      II=I
      IF(I.GT.24) II=I-24
   30 FLHOUR(J)=FLHOUR(J)+FLHR(II,J)
      FLHOUR(J)=FLHOUR(J)/HRS
      FRC=FRC+FLHOUR(J)*FLMO(IMONTH,J)*FLDY(IDAY,J)
      HC=HC+HCWRK(J+2,N)*TVP(J)*FLMO(IMONTH,J)*FLDY(IDAY,J)
     . *FLHOUR(J)*7./DAYS
   40 CONTINUE
      FRC=FRC/4.*7./DAYS
C
      J=HCWRK(2,N)
      ABAR(7,J)=ABAR(7,J) + (HC + HCWRK(10,N) * FRC) *(1.E+6/3.6)
   50 CONTINUE
C
  100 IF (NBRT.EQ.0) GO TO 200
C     CALCULATE HC EMISSIONS FROM SPECIFIED FUEL AND
C     TANK TYPES.  ACCUMULATE IN ABAR ARRAY
C
      DO 110 N=1,NBRT
      J=HCBRT(3,N)
      EX=0.68
      IF (HCBRT(4,N).EQ.2.) EX=0.70
      HC=HCBRT(5,N)*(TVP(J)/(14.7-TVP(J)))**EX*(1.E+6/(3.6*24.*365.))
C
      J=HCBRT(2,N)
      ABAR(7,J)=ABAR(7,J)+HC
  110 CONTINUE
C
  200 IF (NXEVP.EQ.0) GO TO 300
      ICLASS=110
      NTEMP=NPLTS
      NPLTS=1
      LOC1=3
      NSRCE=NABARS
      DO 210 N=1,NXEVP
      DO 210 I=1,3
      SORCE(I,NABARS+N)=HCEVP(I,N)
  210 CONTINUE
      IOPT=1
      IF (JFLAG.EQ.0) READ 2,IOPT
    2 FORMAT(I4)
      GO TO (220,230),IOPT
C
  220 CALL METHA(NXEVP,SORCE,I1,I2,ICLASS)
      GO TO 240
  230 CALL METHC(NXEVP,SORCE,I1,I2,ICLASS)
C
  240 DO 250 N=1,NXEVP
C
C     ACCUMULATE OTHER EVAPORATIVE HC EMISSIONS IN ABAR ARRAY
C
      J=HCEVP(2,N)
      ABAR(7,J)=ABAR(7,J)+SORCE(3,NABARS+N)
  250 CONTINUE
C
      NPLTS=NTEMP
      NSRCE=0
```

```
      LCC1=2
C
  300 IF (NSHS.EQ.0) GO TO 400
      ICLASS=111
      CALL METHB(NSHS,SORCE,I1,I2,ICLASS)
C
  400 IF (NORVHS.EQ.0) GO TO 500
      ICLASS=112
      CALL METHC(NORVHS,SORCE,I1,I2,ICLASS)
C
  500 IF (NMVHAR.EQ.0) GO TO 600
      CALL METHE(NMVHAR,SORCE,VHMLMO,VHMLDY,VHMLHR,I1,I2)
C
  600 IF (NCVHAR.EQ.0) GO TO 700
      CALL METHE(NCVHAR,SORCE,CVABMO,CVABDY,CVABHR,I1,I2)
C
C****EMISSIONS ARE NOW IN MICROGRAMS/SEC
C     FILL ABAR ARRAY WITH AREA GEOMETRIES
C
  700 DO 710 N=1,NABAR
      DO 710 I=1,5
      ABAR(I,N)=ABARGM(I+2,N)
  710 CONTINUE
C
C     FILL ABAR ARRAY WITH THE NON-EVAP HC EMISSION DATA
C
      I1=11
      I2=100
      NSRCE=0
      LCC1=5
      IF (NSHS.EQ.0) GO TO 800
      CALL EMISAR(NSHS,ABAR,I1,I2)
C
  800 IF (NORVHS.EQ.0) GO TO 900
      CALL EMISAR(NORVHS,ABAR,I1,I2)
C
  900 IF (NMVHAR.EQ.0) GO TO 1000
      CALL EMISAR(NMVHAR,ABAR,I1,I2)
C
 1000 IF (NCVHAR.EQ.0) GO TO 1100
      CALL EMISAR(NCVHAR,ABAR,I1,I2)
C
 1100 CONTINUE
      RETURN
      END
```

SUBROUTINE ABLNAR

Purpose:

1. To read from the master source tape all data needed to define airbase non-aircraft line sources.
2. To compute the emission rates due to military and civilian vehicle line and other airbase line activities.

Input:

If the diurnal distribution cards are input, an additional parameter, IMETH, is input here to choose the method of distribution of emissions from those other airbase line activities not using the default of a uniform distribution.

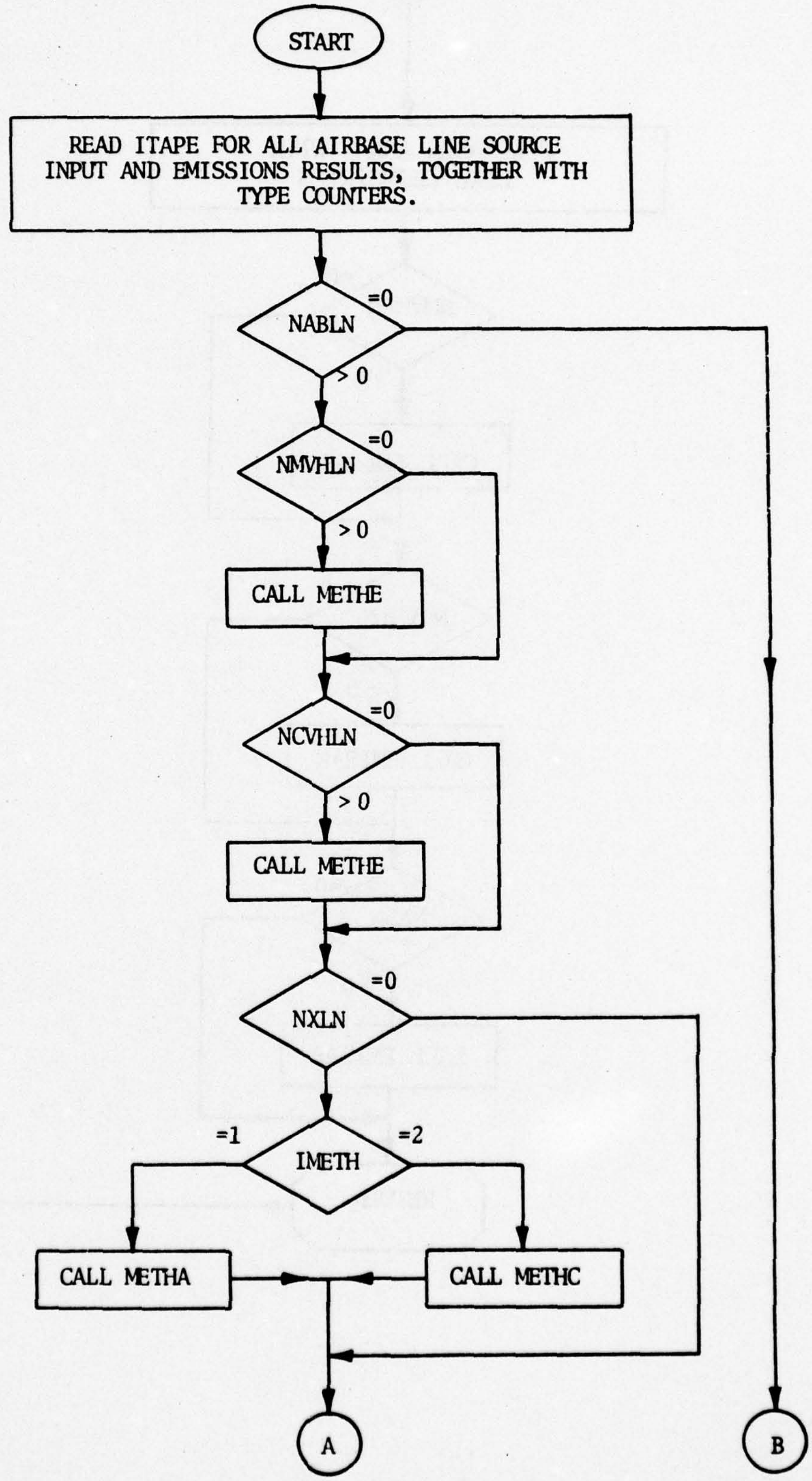
Output:

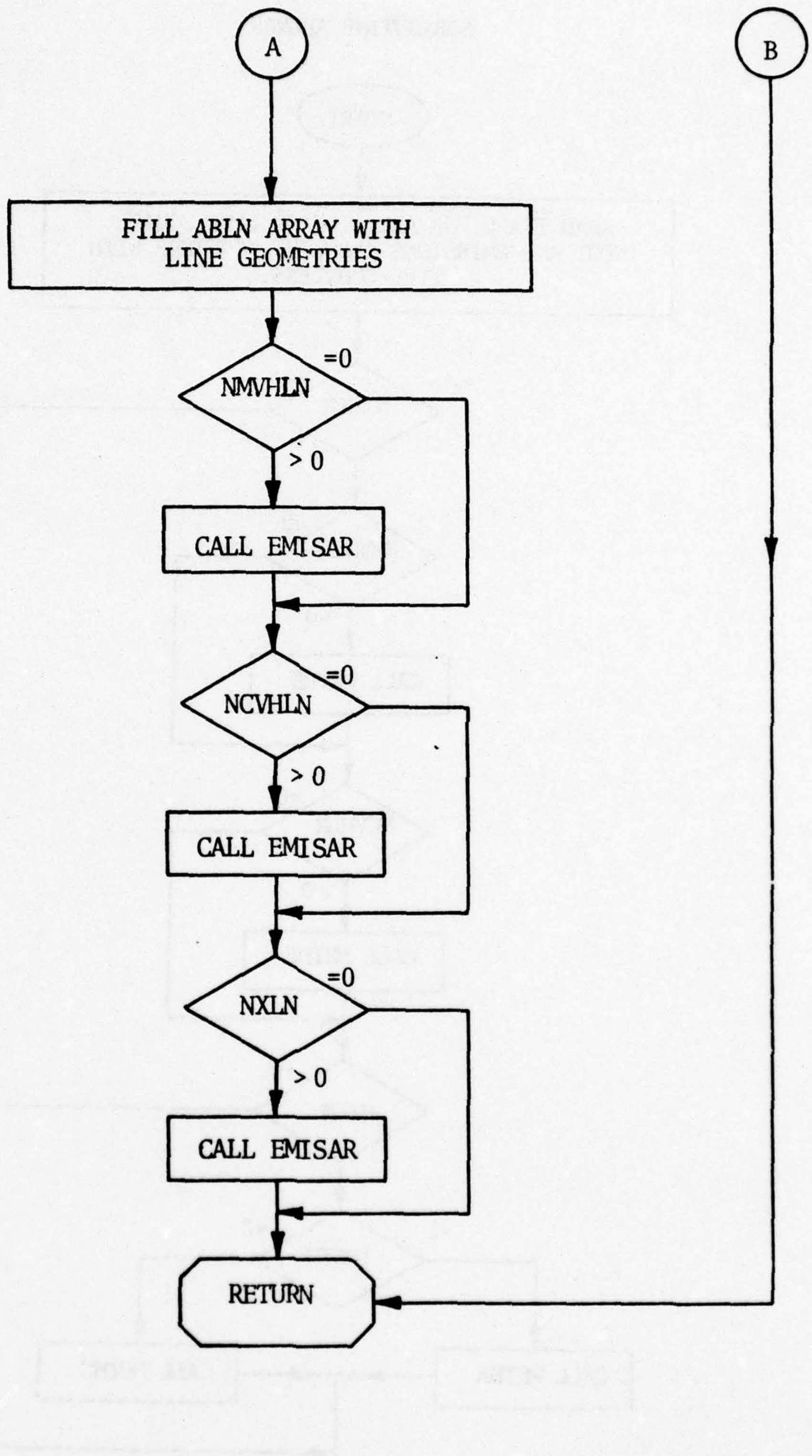
The array, ABLN, is filled with geometry and emission data for non-aircraft line sources.

Subroutines Called:

METHA, METHC, METHE, EMISAR

SUBROUTINE ABLNAR




A
B
FILL ABLN ARRAY WITH LINE GEOMETRIES
NMVHLN
=0
> 0
CALL EMISAR
NCVHLN
=0
> 0
CALL EMISAR
NXLN
=0
> 0
CALL EMISAR
RETURN

```
      SUBROUTINE ABLNAR                                                 ABLNR000
C                                                                       ABLNR001
C     THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL                  ABLNR002
C     AIRBASE LINES                                                     ABLNR003
C        NMVHLN = NO. OF MILITARY LINE ACTIVITIES                       ABLNR004
C        NCVHLN = NO. OF CIVILIAN LINE ACTIVITIES                       ABLNR005
C        NXLN   = NO. OF OTHER AIR BASE LINE ACTIVITIES                 ABLNR006
C                                                                       ABLNR007
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7) ABLNR008
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG          ABLNR009
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)          ABLNR010
     . ,LOC1,LOC2,NGEOM,IPT                                             ABLNR011
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),       ABLNR012
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13), ABLNR013
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1         ABLNR014
      COMMON /SPCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,          ABLNR015
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),         ABLNR016
     .                   ABPT(16,150),ABAR(11,100),ABLN(14,100)         ABLNR017
      LOC1=2                                                            ABLNR018
      LOC2=2                                                            ABLNR019
      I1=17                                                             ABLNR020
      I2=300                                                            ABLNR021
      NGEOM=0                                                           ABLNR022
      IPT=0                                                             ABLNR023
      NSRCE=0                                                           ABLNR024
      READ(ITAPE) NABLN,NTOT,NMVHLN,NCVHLN,NXLN,NABLNS,                 ABLNR025
     .  ((SORGM(I,N),I=1,10),N=1,NABLN),                                ABLNR026
     .  ((SORCE(I,N),I=1,NTOT),N=1,NABLNS)                              ABLNR027
C                                                                       ABLNR028
      IF (NABLN.EQ.0) GO TO 600                                         ABLNR029
      IF (NMVHLN.EQ.0) GO TO 100                                        ABLNR030
C                                                                       ABLNR031
      CALL METHE(NMVHLN,SORCE,VHMLMO,VHMLDY,VHMLHR,I1,I2)               ABLNR032
C                                                                       ABLNR033
  100 IF (NCVHLN.EQ.0) GO TO 200                                        ABLNR034
C                                                                       ABLNR035
      CALL METHE(NCVHLN,SORCE,CVABMO,CVABDY,CVABHR,I1,I2)               ABLNR036
C                                                                       ABLNR037
  200 IF (NXLN.EQ.0) GO TO 300                                          ABLNR038
      ICLASS=117                                                        ABLNR039
C                                                                       ABLNR040
      IMETH=1                                                           ABLNR041
      IF (JFLAG.EQ.0) READ 1,IMETH                                      ABLNR042
    1 FORMAT(I4)                                                        ABLNR043
      GO TO (210,220),IMETH                                             ABLNR044
C                                                                       ABLNR045
  210 CALL METHA(NXLN,SORCE,I1,I2,ICLASS)                               ABLNR046
      GO TO 300                                                         ABLNR047
C                                                                       ABLNR048
  220 CALL METHC(NXLN,SORCE,I1,I2,ICLASS)                               ABLNR049
C                                                                       ABLNR050
C****EMISSIONS ARE NOW IN MICROGRAMS/SEC                                ABLNR051
C     FILL ABLN ARRAY WITH LINE GEOMETRIES                              ABLNR052
C                                                                       ABLNR053
  300 DO 320 N=1,NABLN                                                  ABLNR054
      DO 310 I=1,8                                                      ABLNR055
      ABLN(I,N)=SORGM (I+2,N)                                           ABLNR056
  310 CONTINUE                                                          ABLNR057
      DO 320 I=1,NPLTS                                                  ABLNR058
      ABLN(I+8,N)=0.0                                                   ABLNR059
  320 CONTINUE                                                          ABLNR060
C                                                                       ABLNR061
```

```
C     FILL ABLN ARRAY WITH LINE EMISSION DATA
C
      NSRCE=0
      LOC1=8
      I1=14
      I2=100
      IF (NMVHLN.EQ.0) GO TO 400
      CALL EMISAR(NMVHLN,ABLN,I1,I2)
C
  400 IF (NCVHLN.EQ.0) GO TO 500
      CALL EMISAR(NCVHLN,ABLN,I1,I2)
C
  500 IF (NXLN.EQ.0) GO TO 600
      CALL EMISAR(NXLN,ABLN,I1,I2)
C
  600 CONTINUE
      RETURN
      END
```

SUBROUTINE ABPTAR

Purpose:

1. To read from the master source tape all data needed to define airbase non-aircraft point sources.
2. To compute the emission rates due to training fires, test cells, run-up stands, power plants, incinerators, storage tanks and other airbase point source activities.

Input:

None

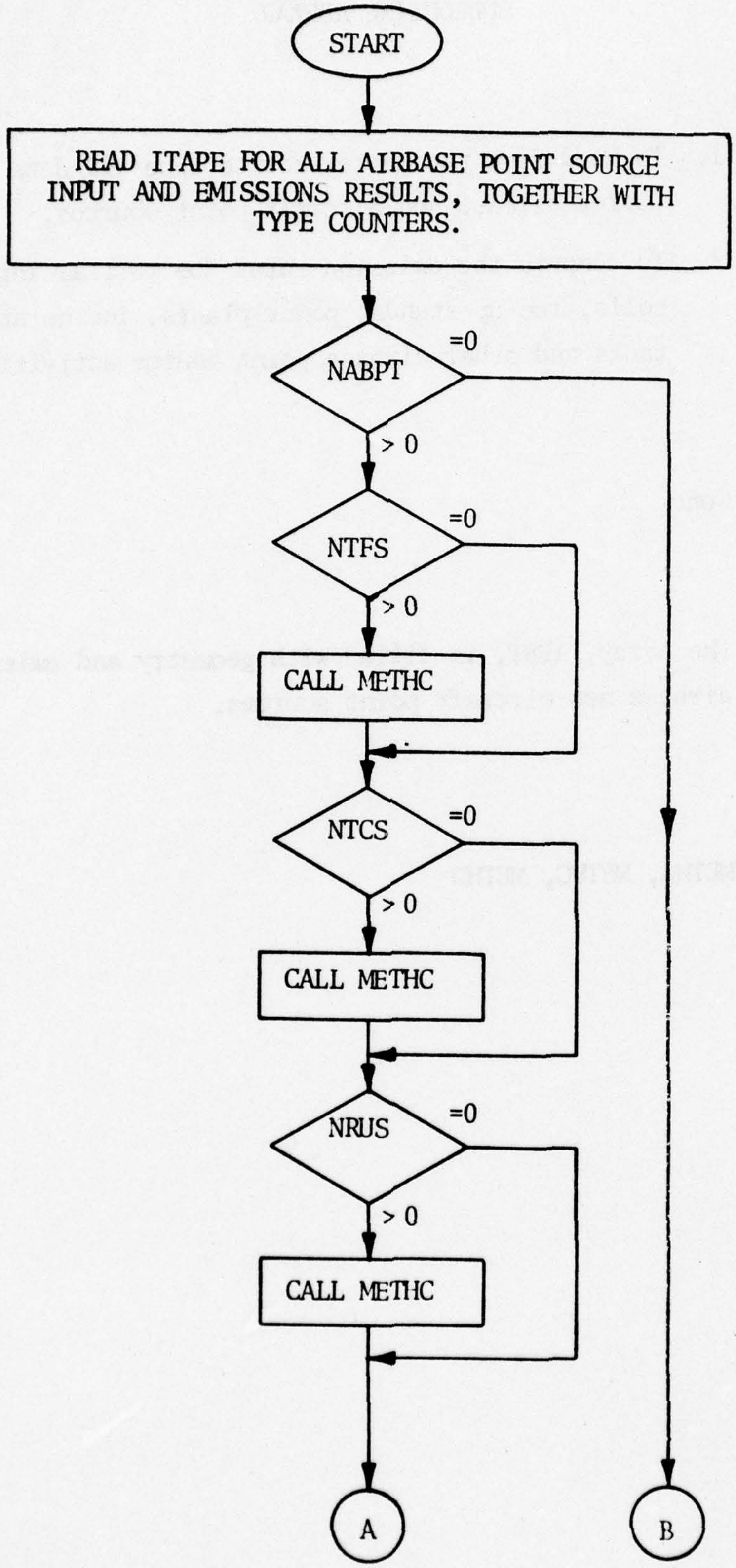
Output:

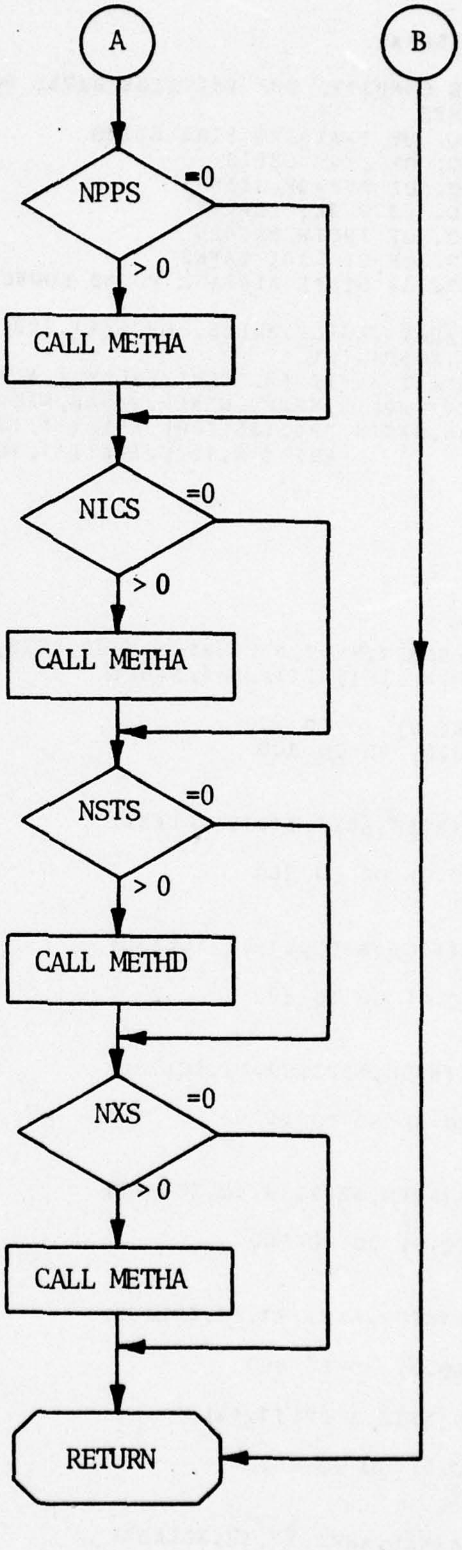
The array, APBT, is filled with geometry and emission data for airbase non-aircraft point sources.

Subroutines Called:

METHA, METHC, METHD

SUBROUTINE ABPTAR




A
B
NPPS
=0
> 0
CALL METHA
NICS
=0
> 0
CALL METHA
NSTS
=0
> 0
CALL METHD
NXS
=0
> 0
CALL METHA
RETURN

```
      SUBROUTINE ABPTAR
C
C     THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL
C     AIRBASE POINTS
C        NTFS = NO. OF TRAINING FIRE SITES
C        NTCS = NO. OF TEST CELLS
C        NRUS = NO. OF RUN-UP STANDS
C        NPPS = NO. OF POWER PLANTS
C        NICS = NO. OF INCINERATORS
C        NSTS = NO. OF STORAGE TANKS
C        NXS  = NO. OF OTHER AIRBASE POINT SOURCES
C
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                   ABPT(16,150),ABAR(11,100),ABLN(14,100)
      LOC1=10
      LOC2=11
      NGEOM=9
      IPT=1
      NSRCE=0
      I1=16
      I2=200
      READ(ITAPE) NABPT,NTOT,NTFS,NTCS,NRUS,NPPS,NICS,NSTS,NXS,
     .  ((SORCE(I,N),I=1,NTOT),N=1,NABPT)
C
      IF (NABPT.EQ.0) GO TO 700
      IF (NTFS.EQ.0) GO TO 100
      ICLASS=101
C
      CALL METHC(NTFS,ABPT,I1,I2,ICLASS)
C
  100 IF (NTCS.EQ.0) GO TO 200
      ICLASS=102
C
      CALL METHC(NTCS,ABPT,I1,I2,ICLASS)
C
  200 IF (NRUS.EQ.0) GO TO 300
      ICLASS=103
C
      CALL METHC(NRUS,ABPT,I1,I2,ICLASS)
C
  300 IF (NPPS.EQ.0) GO TO 400
      ICLASS=104
C
      CALL METHA(NPPS,ABPT,I1,I2,ICLASS)
C
  400 IF (NICS.EQ.0) GO TO 500
      ICLASS=105
C
      CALL METHA(NICS,ABPT,I1,I2,ICLASS)
C
  500 IF (NSTS.EQ.0) GO TO 600
C
      CALL METHD(NSTS,ABPT,I1,I2)
C
  600 IF (NXS.EQ.0) GO TO 700
      ICLASS=107
C
      CALL METHA(NXS,ABPT,I1,I2,ICLASS)
```

```
C
  700 RETURN                                                            ABPTR062
      END                                                               ABPTR063
                                                                        ABPTR064
```

SUBROUTINE ACSRCE

Purpose:

To set up the aircraft source arrays to be used by the dispersion routines for calculating ground level concentrations.

Input:

Basic aircraft data, airbase activity data, points in arrival-departure paths and in training flight paths, meteorological conditions, time period of calculation.

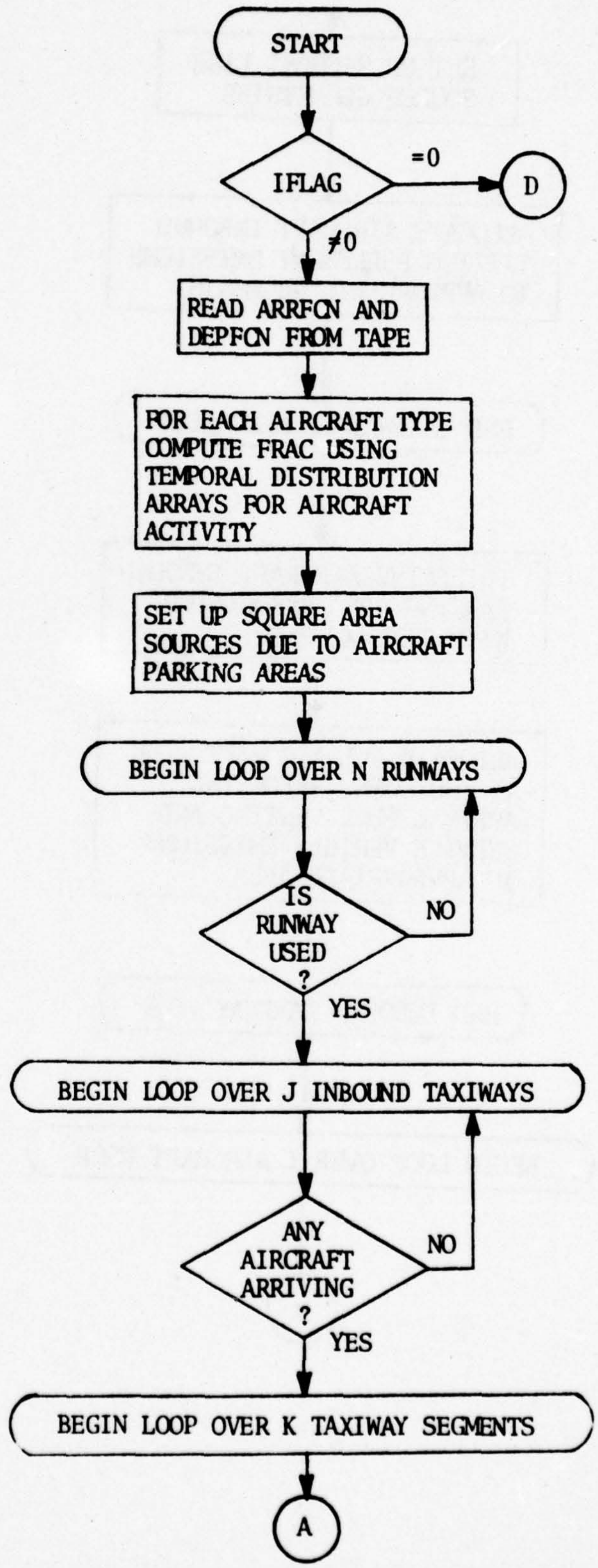
Output:

The arrays ACPT, ACLN and ACAR to contain all source information necessary to calculate dispersion and pollutant concentrations.

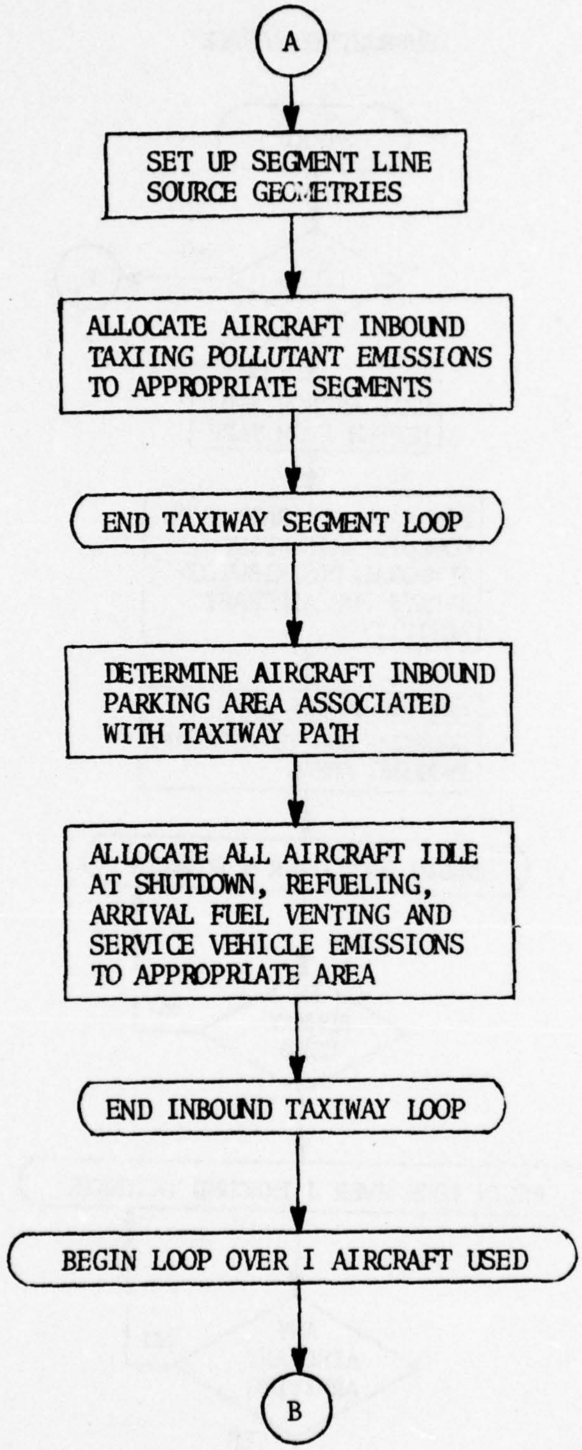
Subroutine Called:

DEPART

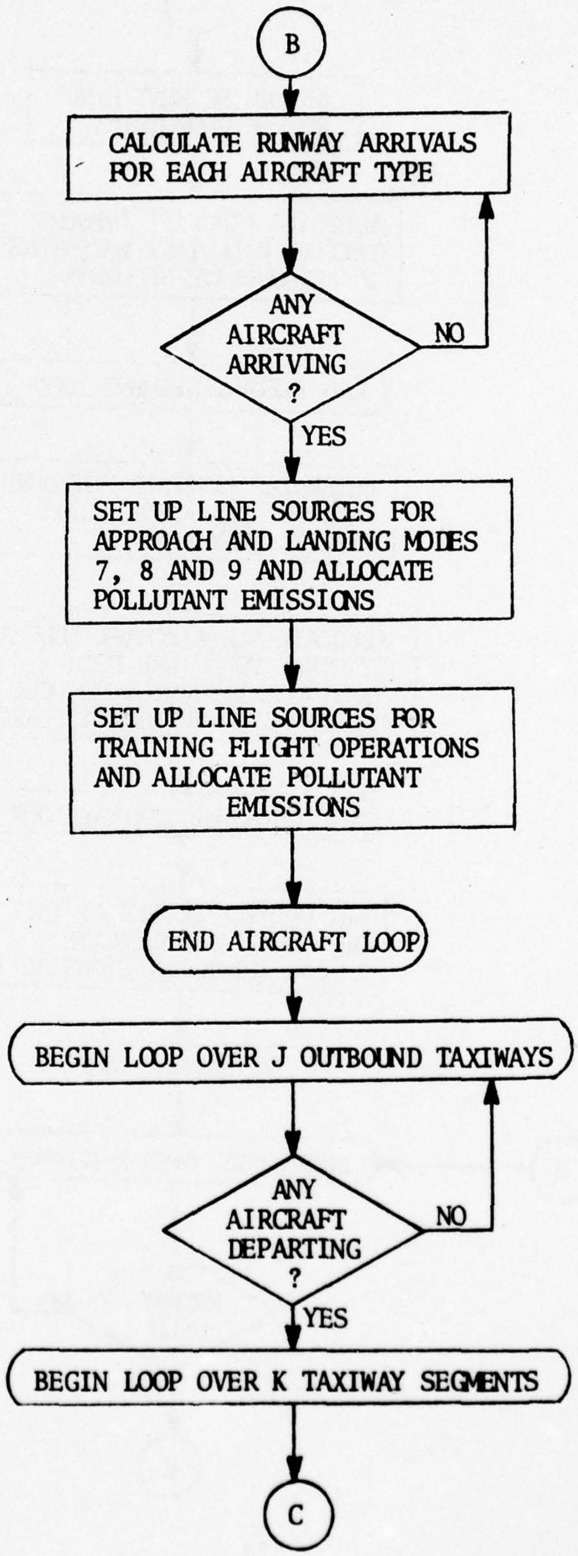
SUBROUTINE ACSRCE



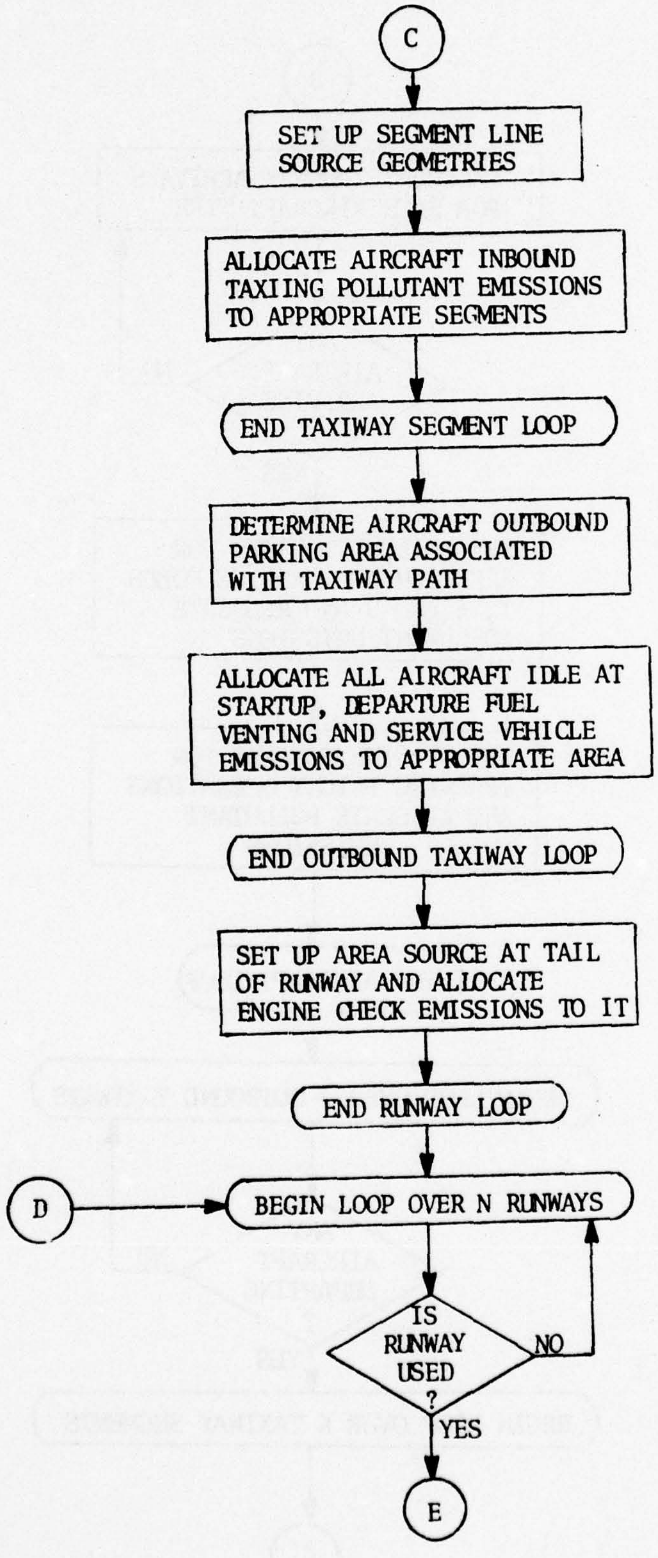
SUBROUTINE ACSRCE (Cont'd.)



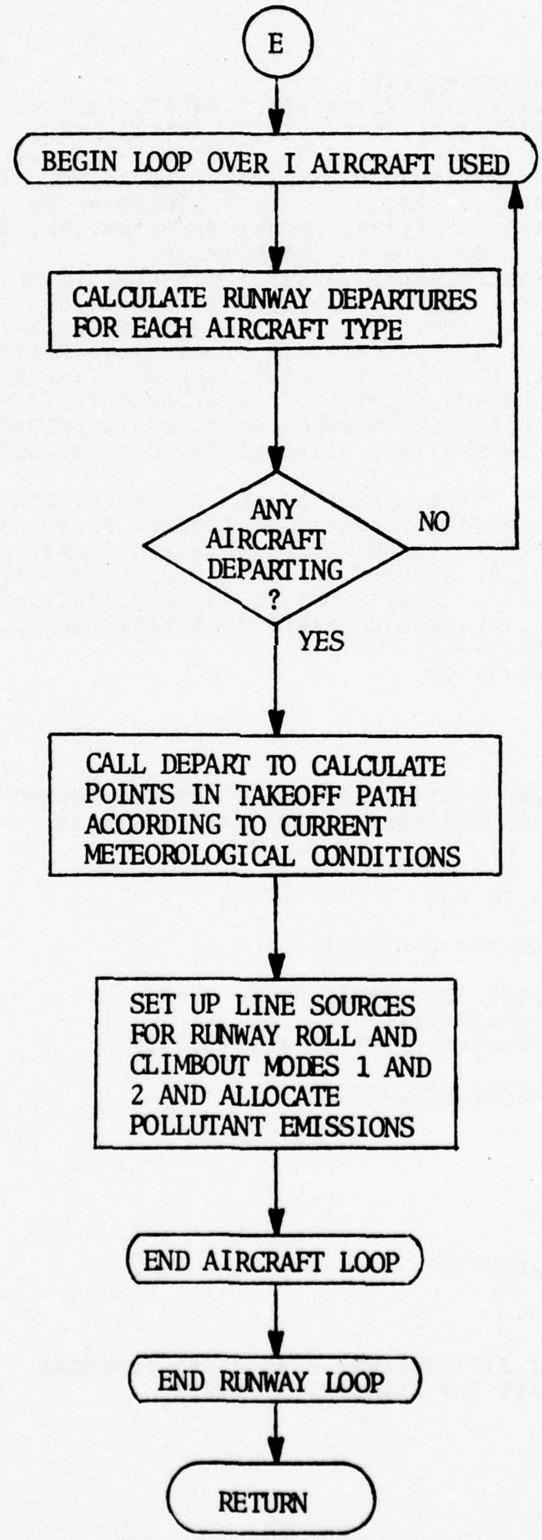
SUBROUTINE ACSRCE (Cont'd.)



SUBROUTINE ACSRCE (Cont'd.)



SUBROUTINE ACSRCE (Cont'd.)



```
      SUBROUTINE ACSRCE
C
C     THIS ROUTINE SETS UP THE AIRCRAFT SOURCE ARRAYS
C     AND ALLOCATES THE POLLUTANT EMISSIONS TO THE
C     APPROPRIATE AREA OR LINE
C
      REAL LNDSPD
      INTEGER  ENGNO
      COMMON /RECPT/ MRECPT,MAXFIL
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),
     . LNDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),
     . COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2)
      COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),
     . ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),
     . IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),
     . IOBSEG(16,8,6),IOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)
      COMMON / MET / WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     1 TEMK
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON /DSTRET/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON /PERIOD/ IMO   ,NODAYS,IDY ,IHR1,IHR2,IFLAG,JFLAG
      DIMENSION IACAR(2,18),FRAC(8),PARFCT(18),APARSQ(6,3),NQ(25)
      XP(XO,YC,W)=YC*SIN(W)+XO
      YP(YO,YC,W)=YC*COS(W)+YO
      DAYS=NODAYS
      NT=NPLTS+5
      IWIND=29+IWD
C
C     AN IFLAG OF 0 MEANS THAT ALL AIRCRAFT SOURCES EXCEPT
C     FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2 REMAIN
C     UNCHANGED
C
      IF(IFLAG.EQ.0) GO TO 69
C
C     READ ARRFCN AND DEPFCN FROM TAPE
C
      IF (IWD.GE.1.AND.IWD.LE.MAXFIL) GO TO 1000
      PRINT 9000,MRECPT,MAXFIL,IWD
 9000 FORMAT(29HCFILE REQUEST ERROR IN ACSRCE,3I5)
      GO TO 1040
 1000 IF (MRECPT-IWD) 1010,1030,1020
 1010 READ (30)
      MRECPT=MRECPT+1
      GO TO 1000
 1020 REWIND 30
      MRECPT=1
      GO TO 1000
 1030 READ (30) ARRFCN,DEPFCN
      MRECPT=MRECPT+1
 1040 CONTINUE
C
C     FOR EACH AIRCRAFT TYPE COMPUTE FRAC USING TEMPORAL
C     DISTRIBUTION ARRAYS FOR AIRCRAFT ACTIVITY
```

```
C
      NHI=IHR2
      IF(IHR1.GT.IHR2) NHI=24+IHR2
      HRS=NHI-IHR1+1
      DO 5 I=1,NACTYP
      HRFRC=0.
      DO 4 JJ=IHR1,NHI
      J=JJ
      IF(JJ.GT.24) J=JJ-24
    4 HRFRC=HRFRC+ACHR(J,I)
      HRFRC=HRFRC/HRS
      FRAC(I)=ACMO(IMO,I)*ACDY(IDY,I)*HRFRC*7.0/DAYS*(1.E+6/3.6)
    5 CONTINUE
    8 NACPT=0
      NB=0
      NC=0
      NZ=0
C
C     SET UP SQUARE AREA SOURCES DUE TO AIRCRAFT PARKING AREAS
C
      DO 1  L=1,NPKAR
      NSQ=NPASQ(L)
      SPARSQ=0.0
      DO 2 J=1,NSQ
      NB=NB+1
      ACAR(1,NB)=PAREA(L,J,1)
      ACAR(2,NB)=PAREA(L,J,2)
      ACAR(3,NB)=ACLNDZ/2.
      ACAR(4,NB)=PAREA(L,J,3) *1000.
      APARSQ(L,J)=ACAR(4,NB) ** 2
      SPARSQ = SPARSQ + APARSQ(L,J)
      ACAR(5,NB)=ACLNDZ
      IACAR(1,NB)=IDPRKA(L)
    2 IACAR(2,NB)=NSQ
      DO 91 J=1,NSQ
      NZ=NZ+1
   91 PAFFCT(NZ) = APARSQ(L,J) / SPARSQ
    1 CONTINUE
C
      DO 93 I=1,NLSEGS
   93 NQ(I)=0
      NPKSRC=NB
      DO 3  L=1,NPKSRC
      DO 3  K=6,NT
      HRACAR(K-5,L)=0.0
    3 ACAR(K,L)=0.0
      TVP= EXP(ALPHA(2)-BETA(2)/TEMK)
C
C     BEGIN LOOP OVER N RUNWAYS
C
      DO 10 N=1,NRNWYS
C
C     IS RUNWAY USED WITH THIS WIND DIRECTION?
C
      IF (IUSWD(IWD,N).EQ.0) GO TO 10
      THETA=RNWY(7,N)
      XO=0.25*DISRNW(N)*SIN(THETA)+RNWY(2,N)
      YO=0.25*DISRNW(N)*COS(THETA)+RNWY(3,N)
      NTT=NIBTT(N)
      IF(NTT.EQ.0) GO TO 50
C
C     BEGIN LOOP OVER J INBOUND TAXIWAYS
```

```
C
      DO 11 J=1,NTT
C
C     ANY AIRCRAFT ARRIVING ON THIS RUNWAY?
C
      DO 7 I=1,NACTYP
      IF(TTARFR(J,I,N)*ARRFCN(23,I,N).GT.0.0) GO TO 701
    7 CONTINUE
      GO TO 11
  701 NSGLNS = NIBSEG (J,N)
C
C     BEGIN LOOP OVER K TAXIWAY SEGMENTS
C
      DO 12 K=1,NSGLNS
C
C     SET UP SEGMENT LINE SOURCE GEOMETRIES
C
      JJ = IIBSEG(K,J,N)
      IF(NQ(JJ).NE.0) GO TO 130
      NC=NC+1
      NQ(JJ)=NC
      DO 121 L=1,12
  121 ACLN(L,NC)=ACLNSG(L,JJ)
      ACLN(9,NC)=1.0
      ACLN(10,NC)=1.0
C
C     ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS
C     TO APPROPRIATE SEGMENTS
C
      DO 13 L=1,NPLTS
      LL=L+12
 13   ACLN(LL,NC)=0.0
  130 ND=NQ(JJ)
      DO 14 I=1,NACTYP
      AA=ENGNO(I ,1)
      IF(IEGFLG.GT.0) AA=ENGNO(I ,2)
      ARR=TTARFR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)
      IF(ARR.LE.0.0)GO TO 14
      TIME=ACLN(11,ND)/(TXISPD(I )*ACLNSG(9,JJ))
      FRC=AA*ARR*TIME*FRAC(I )
      DO 15 L=1,NPLTS
      KK=L+12
 15   ACLN(KK,ND)=ACLN(KK,ND)+FRC*ACEMFC(I ,2,L)
 14   CONTINUE
  12  CONTINUE
C
C     END TAXIWAY SEGMENT LOOP
C
C
C     DETERMINE AIRCRAFT INBOUND PARKING AREA
C     ASSOCIATED WITH TAXIWAY PATH
C
      DO 16 I=1,NPKSRC
      II=I
      IDPK=IACAR(1,I)
      IF(IDPK.EQ.IDIBPA(J,N))GO TO 17
  16  CONTINUE
      PRINT 18, IDIBPA(J,N),J,N
 18   FORMAT ('0INBOUND PARKING AREA 'I3,'OF TAXIWAY='I3,'; RUNWAY='I3,'
     1 IS NOT CONSISTANT WITH PARKING AREA ID NUMBERS')
      STOP
 17   CONTINUE
```

```
C
C     AILCCATE ALI AIRCRAFT IDLE AT SHUTDOWN, REFUELING,
C     APRIVAL FUEL VENTING AND SERVICE VEHICLE EMISSIONS
C     TC APPROPRIATE AREA
C
      NSQ=IACAR(2,II)
      DC  19 I=1,NACTYP
      ARR=TTARRR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)
      IF(ARR.LE.0.0) GO TO 19
      AA=ENGNO(I,1)
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)
      TIME=SHTDNT(I)/60.
      FRC=AA*ARR*TIME*FRAC(I)
      TVP=EXP(ALPHA(JES1(I)) - BETA(JES1(I)) / TEMK)
      DC 20 L=1,NSQ
      JJ=II+L-1
      DO 21 K=1,NPLTS
      KK=K+5
      ACAR(KK,JJ)=ACAR(KK,JJ)+FRC*ACEMFC(I,1,K) * PARFCT(JJ)
      ACAR(KK,JJ)=ACAR(KK,JJ) + (ARSVEM(K,I,1) + ARSVEM(K,I,2) +
     .ARSVEM(K,I,3) + ARSVEM(K,I,4)+ARSVEM(K,I,5)) * ARR * FRAC(I)
     . * PARFCT(JJ)
      IF(K.EQ.2) ACAR(KK,JJ)=ACAR(KK,JJ)+(0.3*TVP*ACFUEL(I)*0.5
     1/1000. + ACSPIL(I) + ARFLVT(I)) * ARR * FLDENS(JES1(I)) * FRAC(I)
     . * PARFCT(JJ)
   21 CONTINUE
   20 CONTINUE
   19 CONTINUE
   11 CONTINUE
C
C     END INBOUND TAXIWAY LOOP
C
C
C     BEGIN LOOP OVER I AIRCRAFT USED
C
      DC 30 I=1,NACTYP
C
C     CALCULATE RUNWAY ARRIVALS FOR EACH AIRCRAFT TYPE
C
      ARR=ARRFCN(23,I,N)*ANNARR(I)
C
C     ANY AIRCRAFT ARRIVING?
C
      IF(ARR.LE.0.0) GO TO 30
C
C     SET UP LINE SOURCES FOR APPROACH AND LANDING MODES 7, 8 AND 9
C     AND AILCCATE POLLUTANT EMISSIONS
C
      AA=ENGNC(I,1)
      DC 31 J=1,3
      DO 32 K=1,3
      KK=K+NC
      JK=6*K-6+J
      ACLN(J,KK)=ARRFCN(JK,I,N)
      ACLN(J+5,KK)=ARRFCN(JK+6,I,N)
   32 CCNTINUE
      JJ=NC+J
      JK=6*J-2
      ACLN(4,JJ)=ARRFCN(24,I,N)
      ACLN(5,JJ)=DEPFCN(24,I,N)
      ACLN(09,JJ)=ARRFCN(JK,I,N)
      ACLN(10,JJ)=ARRFCN(JK+6,I,N)
```

```
      ACLN(11,JJ)=ARRFCN(JK+1,I,N)
      ACLN(12,JJ)=ARRFCN(JK+2,I,N)
      JMODE=J+6
      DO 33 K=1,NPLTS
      KK=K+12
      ACLN(KK,JJ)=AA*ACEMFC(I,JMODE,K)*ARR*ARRFCN(JK+2,I,N)*FRAC(I)
   33 CONTINUE
   31 CONTINUE
      NC=NC+3
C
C     SET UP LINE SOURCES FOR TRAINING FLIGHT OPERATIONS
C     AND ALLOCATE POLLUTANT EMISSIONS
C
      IF(ANNTGO(I).LE.0.0) GO TO 30
      NC=NC+1
      ACLN(1,NC)=XP(XO,TGO(1,1,I),THETA)
      ACLN(2,NC)=YP(YO,TGO(1,1,I),THETA)
      ACLN(6,NC)=XP(XO,TGO(1,2,I),THETA)
      ACLN(7,NC)=YP(YO,TGO(1,2,I),THETA)
      ACLN(1,NC+1)=ACLN(6,NC)
      ACLN(2,NC+1)=ACLN(7,NC)
      ACLN(6,NC+1)=XO
      ACLN(7,NC+1)=YO
      ACLN(1,NC+2)=XO
      ACLN(2,NC+2)=YO
      ACLN(6,NC+2)=XP(XO,0.3048,THETA)
      ACLN(7,NC+2)=YP(YO,0.3048,THETA)
      ACLN(1,NC+3)=ACLN(6,NC+2)
      ACLN(2,NC+3)=ACLN(7,NC+2)
      ACLN(6,NC+3)=XP(XO,TGO(1,3,I),THETA)
      ACLN(7,NC+3)=YP(YO,TGO(1,3,I),THETA)
      ACLN(1,NC+4)=ACLN(6,NC+3)
      ACLN(2,NC+4)=ACLN(7,NC+3)
      ACLN(6,NC+4)=XP(XO,TGO(1,4,I),THETA)
      ACLN(7,NC+4)=YP(YO,TGO(1,4,I),THETA)
      ACLN(3,NC)=APPHT*1000.
      ACLN(8,NC)=APPHT2(I)*1000.
      ACLN(3,NC+1)=APPHT2(I)*1000.
      ACLN(8,NC+1)=ACLNDZ/2.
      ACLN(3,NC+2)=ACLNDZ/2.
      ACLN(8,NC+2)=ACLNDZ/2.
      ACLN(3,NC+3)=ACLNDZ/2.
      ACLN(8,NC+3)=COHT1(I)*1000.
      ACLN(3,NC+4)=COHT1(I)*1000.
      ACLN(8,NC+4)=CLMBHT*1000.
      ACLN(09,NC)=APSPD1(I)
      ACLN(10,NC)=APSPD2(I)
      ACLN(11,NC)=TGO(2,1,I)
      ACLN(12,NC)=TGO(3,1,I)
      ACLN(09,NC+1)=APSPD2(I)
      ACLN(10,NC+1)=LNDSPD(I)
      ACLN(11,NC+1)=TGO(2,2,I)
      ACLN(12,NC+1)=TGO(3,2,I)
      ACLN(09,NC+2)=LNDSPD(I)*1.3
      ACLN(10,NC+2)=TOSPD(I)*0.7
      ACLN(11,NC+2)=0.3048
      ACLN(12,NC+2)=2.0*0.3048/(1.3*LNDSPD(I)+0.7*TOSPD(I))
      ACLN(09,NC+3)=TOSPD(I)
      ACLN(10,NC+3)=COSPD1(I)
      ACLN(11,NC+3)=TGO(2,3,I)
      ACLN(12,NC+3)=TGO(3,3,I)
      ACLN(09,NC+4)=COSPD1(I)
```

```
      ACLN(10,NC+4)=COSPD2(I)
      ACLN(11,NC+4)=TGO(2,4,I)
      ACLN(12,NC+4)=TGO(3,4,I)
      DO 45 J=1,5
      JJ=NC+J-1
      ACLN(4,JJ)=ARRFCN(24,I,N)
      ACLN(5,JJ)=DEPFCN(24,I,N)
      GO TO (34,35,41,36,37),J
   34 KD=7
      GO TO 38
   35 KD=8
      GO TO 38
   36 KD=5
      GO TO 38
   37 KD=6
   38 DO 39 K=1,NPLTS
      KK=K+12
   39 ACLN(KK,JJ)=ANNTGO(I)*ACEMFC(I,KD,K)*ARRFCN(23,I,N)*ACLN(12,JJ)*
     1FRAC(I)*AA
      GO TO 45
   41 DO 42 K=1,NPLTS
      KK=K+12
   42 ACLN(KK,JJ)=AA*(0.3*ACEMFC(I,9,K)+0.7*ACEMFC(I,4,K))*
     1 ANNTGO(I)*ARRFCN(23,I,N)*ACLN(12,JJ)*FRAC(I)
   45 CONTINUE
      NC=NC+4
   30 CONTINUE
C
C     END AIRCRAFT LOOP
C
   50 NTT=NOBTT(N)
      IF(NTT.EQ.0) GO TO 10
C
C     BEGIN LOOP OVER J OUTBOUND TAXIWAYS
C
      DO 51 J=1,NTT
C
C     ANY AIRCRAFT DEPARTING ON THIS TAXIWAY?
C
      DO 6 I=1,NACTYP
      IF(TTDPFR(J,I,N)*DEPFCN(23,I,N).GT.0.0) GO TO 601
    6 CONTINUE
      GO TO 51
  601 NSGLNS=NOBSEG(J,N)
C
C     BEGIN LOOP OVER K TAXIWAY SEGMENTS
C
      DO 52 K=1,NSGLNS
C
C     SET UP SEGMENT LINE SOURCE GEOMETRIES
C
      JJ=IOBSEG(K,J,N)
      IF(NQ(JJ).NE.0) GO TO 131
      NC=NC+1
      NQ(JJ)=NC
      DO 122 L=1,12
  122 ACLN(L,NC)=ACLNSG(L,JJ)
      ACLN(9,NC)=1.0
      ACLN(10,NC)=1.0
C
C     ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS
C     TO APPROPRIATE SEGMENTS
```

```
C
      DO 53 L=1,NPLTS
      LL=L+12
   53 ACLN(LL,NC)=0.0
  131 ND=NC(JJ)
      DO 54 I=1,NACTYP
      DEP=TTDPFR(J,I,N)*DEPFCN(23,I,N)*ANNDEP(I)
      IF(DEP.LE.0.0) GO TO 54
      AA=ENGNO(I,1)
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)
      TIME= ACLN(11,ND) / TXISPD(I)
      FRC= AA* DEP*TIME*FRAC(I)
      DO 55 L=1,NPLTS
      KK=L+12
   55 ACLN(KK,ND)= ACLN(KK,ND)+FRC*ACEMFC(I,2,L)
   54 CONTINUE
   52 CONTINUE
C
C     END TAXIWAY SEGMENT LOOP
C
C
C     DETERMINE AIRCRAFT OUTBOUND PARKING AREA ASSOCIATED
C     WITH TAXIWAY PATH
C
      DO 56 I=1,NPKSRC
      II =I
      IDPK=IACAR(1,I)
      IF(IDPK.EQ.IDOBPA(J,N)) GO TO 58
   56 CONTINUE
      PRINT 57,IDOBPA(J,N),J,N
   57 FORMAT(22H0OUTBOUND PARKING AREA,I3,11H OF TAXIWAY,I3,8H, RUNWAY,
     . I3,47H IS NOT CONSISTENT WITH PARKING AREA ID NUMBERS)
      STOP
C
C     ALLOCATE ALL AIRCRAFT IDLE AT STARTUP, DEPARTURE FUEL
C     VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA
C
   58 NSQ=IACAR(2,II )
      DO 59 I=1,NACTYP
      DEP=TTDPFR(J,I,N)*DEPFCN(23,I,N)*ANNDEP(I)
      IF(DEP.EQ.0.0) GO TO 59
      AA=ENGNO(I,1)
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)
      TIME=SRTUPT(I)/60.
      FRC = AA* DEP* TIME * FRAC(I)
      TVP=EXP(ALPHA(JES1(I)) - BETA(JES1(I)) / TEMK)
      DO 60 L=1,NSQ
      JJ =II +L-1
      DO 61 K=1,NPLTS
      KK=K+5
      ACAR(KK,JJ) = ACAR(KK,JJ) + ((FRC * ACEMFC(I,1,K)) +
     . ((DPSVEM(K,I,1) + DPSVEM(K,I,2) + DPSVEM(K,I,3) + DPSVEM(K,I,4)
     . + DPSVEM(K,I,5)) * DEP * FRAC(I))) * PARFCT(JJ)
      IF (K.EQ.2) ACAR(KK,JJ) = ACAR(KK,JJ) + DPFLVT(I) * DEP * FLDENS(
     . JES1(I)) * FRAC(I) * PARFCT(JJ)
   61 CONTINUE
   60 CONTINUE
   59 CONTINUE
   51 CONTINUE
C
C     END OUTBOUND TAXIWAY LOOP
C
```

```
      NB=NB+1
C
C     SET UP AREA SOURCE AT TAIL OF RUNWAY AND ALLOCATE
C     ENGINE CHECK EMISSIONS TO IT
C
      ACAR(1,NB)=RNWY(2,N)-.05 * SIN(THETA)
      ACAR(2,NB)=RNWY(3,N)-.05*COS(THETA)
      ACAR(3,NB)= ACLNDZ/2.
      ACAR(4,NB)= 100.0
      ACAR(5,NB)= ACLNDZ
      DO 65 K=1,NPLTS
      KK=K+5
   65 ACAR(KK,NB)=0.0
      DO 66 I=1,NACTYP
      DEP=DEPFCN(23,I,N)*ANNDEP(I)
      IF(DEP.EQ.0.0) GO TO 66
      AA=ENGNO(I,1)
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)
      TIME= EGCHKT(I)/60.
      FRC= TIME *DEP*AA*FRAC(I)
      DO 67 K=1,NPLTS
      KK=K+5
   67 ACAR(KK,NB)=ACAR(KK,NB)+ FRC* ACEMFC(I,3,K)
   66 CONTINUE
   10 CONTINUE
C
C     END RUNWAY LOOP
C
      NACAR=NB
      NC1=NC
   69 NC=NC1
C
C     BEGIN LOOP OVER N RUNWAYS
C
      DO 79 N=1,NRNWYS
C
C     IS RUNWAY USED WITH THIS WIND DIRECTION?
C
      IF(IUSWD(IWD,N).EQ.0) GO TO 79
C
C     BEGIN LOOP OVER I AIRCRAFT USED
C
      DO 70 I=1,NACTYP
C
C     CALCULATE RUNWAY DEPARTURES FOR EACH AIRCRAFT TYPE
C
      DEP=DEPFCN(23,I,N)*ANNDEP(I)
C
C     ANY AIRCRAFT DEPARTING FROM THIS RUNWAY?
C
      IF(DEP.EQ.0.0) GO TO 70
C
C     CALL DEPART TO CALCULATE POINTS IN TAKEOFF PATH ACCORDING
C     TO CURRENT METEOROLOGICAL CONDITIONS
C
      CALL DEPART (N,I)
      AA=ENGNO(I ,1)
C
C     SET UP LINE SOURCES FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2
C     AND ALLOCATE POLLUTANT EMISSIONS
C
      DO 71 J=1,3
```

```
      DO 72 K=1,3
      KK=K+NC
      JK=6*K-6+J
      ACLN(J,KK)=DEPFCN(JK,I,N)
   72 ACLN(J+5,KK)=DEPFCN(JK+6,I,N)
      JJ=NC+J
      JK=6*J-2
      ACLN(4,JJ)=ARRFCN(24,I,N)
      ACLN(5,JJ)=DEPFCN(24,I,N)
      ACLN(09,JJ)=DEPFCN(JK,I,N)
      ACLN(10,JJ)=DEPFCN(JK+6,I,N)
      ACLN(11,JJ)=DEPFCN(JK+1,I,N)
      ACLN(12,JJ)=DEPFCN(JK+2,I,N)
      JMODE=J+3
      DO 73 K=1,NPLTS
      KK=K+12
      ACLN(KK,JJ)=AA*ACEMFC(I,JMODE,K)*DEP*DEPFCN(JK+2,I,N)*FRAC(I)
   73 CONTINUE
   71 CONTINUE
      NC=NC+3
   70 CONTINUE
C
C     END AIRCRAFT LOOP
C
   79 CONTINUE
C
C     END RUNWAY LOOP
C
      NACLN=NC
      RETURN
      END
```

FUNCTION AINE

Purpose:

1. To translate the line and receptor coordinates to an x-axis along the wind vector, placing the origin of the line at its low end.

2. To set up the necessary parameters for the CAVL and QMOD routines.

3. To determine the concentration due to the given line.

Input:

The current wind direction and speed, and the receptor and line source data.

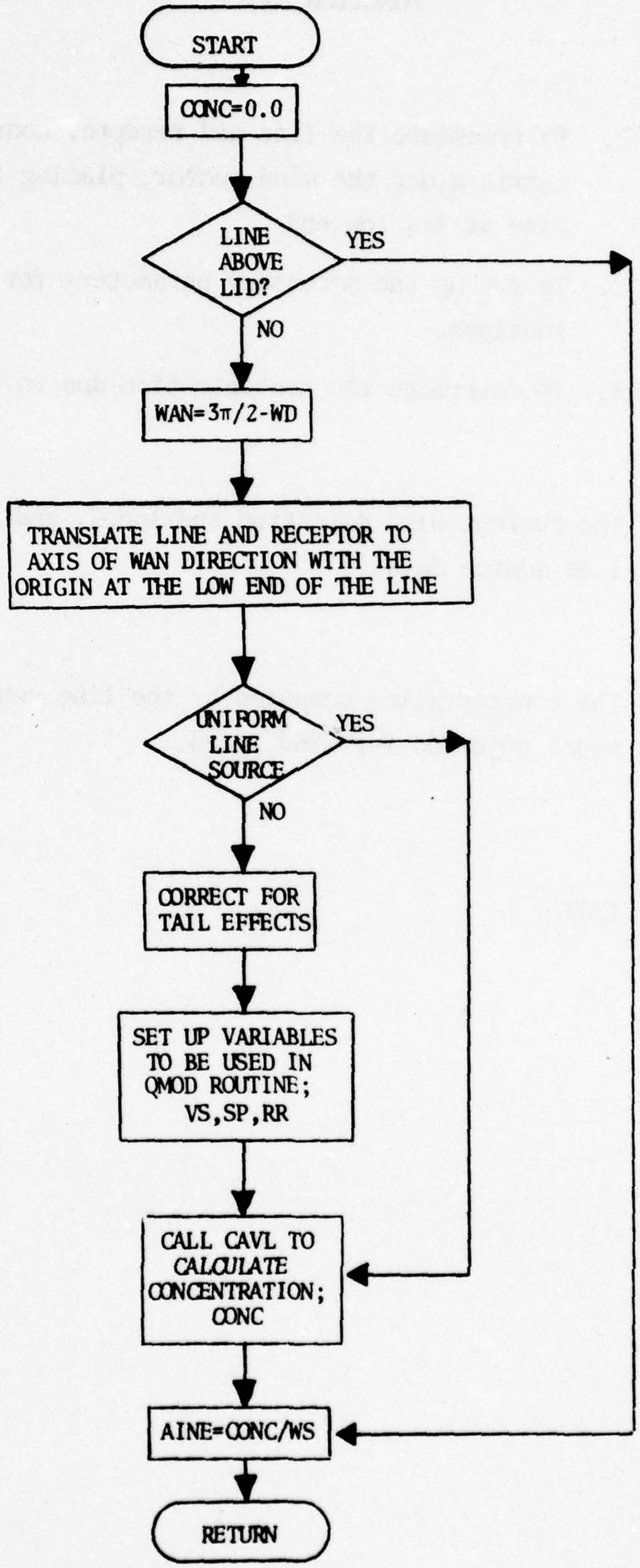
Output:

The concentration computed by the line source diffusion model adjusted for wind speed.

Subroutines Called:

CAVL

FUNCTION AINE(WD)



```
      FUNCTION AINE (WD)
C
C   THIS FUNCTION TRANSLATES THE LINE AND RECEPTOR COORDINATES TO AN
C   X-AXIS ALONG THE WIND VECTOR, PLACING THE ORIGIN OF THE LINE AT
C   ITS LOW END.  THE VEHICLE MOVES FROM (X1,Y1,Z1) TO (X2,Y2,Z2)
C
      COMMON /MET/ WS,WSMPH,IWS,WX,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,
     . V1,V2,DL,TIME,EMIS(6),NPOL
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,
     . WS2,WSC,RR,SP,XST,YST,ZST,XND,YND,ZND
      DATA PI32/4.7123890/
C
      CONC = 0.
C
C     IF LINE IS ABOVE LID, DO NOT CALCULATE CONC
C
      IF(ZW1.GE.HLID-.5) GO TO 60
C
C     TRANSLATE LINE AND RECEPTOR TO AXIS OF WAN DIRECTION
C
      WAN=PI32-WD
      CSAN=COS(WAN)
      SNAN=SIN(WAN)
      XW2=(X2-X1)*CSAN+(Y2-Y1)*SNAN
      YW2=(X1-X2)*SNAN+(Y2-Y1)*CSAN
      XR = RECEP(1,IRECEP) * 1000.
      YR = RECEP(2,IRECEP) * 1000.
      ZST=ZW1
      ZND=ZW2
      IF(Z1.LE.Z2) GO TO 5
      XW2=-XW2
      XST=XW2
      YW2=-YW2
      YST=YW2
      XND=0.0
      YND=0.0
      XRCP=(XR-X2)*CSAN+(YR-Y2)*SNAN
      YRCP=(X2-XR)*SNAN+(YR-Y2)*CSAN
      GO TO 8
    5 CONTINUE
      XST=0.0
      YST=0.0
      XND=XW2
      YND=YW2
      XRCP=(XR-X1)*CSAN+(YR-Y1)*SNAN
      YRCP=(X1-XR)*SNAN+(YR-Y1)*CSAN
    8 CONTINUE
      ZRCP = 2.
C
C     IS THIS A UNIFORM LINE SOURCE
C
   50 IF(IAD.EQ.0) GO TO 500
C
C     CORRECT FOR TAIL EFFECTS IF ARRIVAL OR DEPARTURE
C
      CSA = -XW2 / DL
      WSC = 2 * WS * CSA
      EXT = TAIL / DL
      DX = XW2 * EXT
```

```
      DY = YW2 * EXT                                                    AINF0062
      XW2 = XW2 + DX                                                    AINE0063
      YW2 = YW2 + DY                                                    AINE0064
      VS = TAIL / TIME                                                  AINE0065
      W1 = V1 + VS                                                      AINE0066
      W2 = V2 + VS                                                      AINE0067
      YY1 = SQRT(WS2 + W1 * (W1 + WSC))                                 AINE0068
      YY2 = SQRT(WS2 + W2 * (W2 + WSC))                                 AINE0069
      SP = YY2                                                          AINE0070
      ARG = (YY2+ W2 + WSC/2.) / (YY1+ W1 + WSC/2.)                     AINE0071
      G =YY2 -YY1 - WSC/2. * ALOG(ARG)                                  AINE0072
      RR = A / G                                                        AINE0073
      IF (Z1.NE.Z2.AND.IAD.EQ.1) GO TO 500                              AINE0074
      XRCP = XRCP + DX                                                  AINE0075
      YRCP = YRCP + DY                                                  AINE0076
C                                                                       AINE0077
C   CALCULATE THE CONCENTRATION DUE TO THIS LINE                        AINE0078
C                                                                       AINE0079
  500 CONC=CAVL(XRCP,YRCP,ZRCP)                                         AINE0080
   60 AINE = CONC / WS                                                  AINE0081
      RETURN                                                            AINE0082
      END                                                               AINE0083
```

BLOCK DATA

Purpose:

To initialize data in common blocks.

Input:

None

Output:

None

```
      BLOCK DATA
C
C   INITIALIZE DATA IN COMMON BLOCKS FOR SHORT TERM MODEL
C
      REAL*8 POLNAM
C
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR
      COMMON /DEFALT/ ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,B,V12,VS,
     . WS2,WSC,RR,SP,AA1,AA2,AA3,AA4,AA5,AA6
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE
      COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      COMMON /TITL/ POLNAM( 6),TITLE1(20),IPCHOS( 6),NXPOL,IP
      COMMON /WNDPRO/ XP(6)
C
C     ************************DATA STATEMENTS**************************
C
      DATA XW1, YW1, TAIL / 0.0, 0.0, 140. /
      DATA XP / 0.2, 0.2, 0.2, 0.3, 0.4, 0.4 /
      DATA ALPHA / 11.70365, 11.10675, 12.42382, 12.68789, 13.687,
     . 13.038, 13.024 /
      DATA BETA / 2868.54, 3129.5187, 3276.8848, 5108.4194,5329.139,
     . 4789.301, 4782.209 /
      DATA FLDENS / 0.695, 0.773, 0.693, 0.842, 0.824, 0.807, 0.807 /
      DATA ACLNDY, ACLNDZ / 20.0,8.0 /
      DATA ITAPE, IONCE / 21, 0 /
      DATA POLNAM / 8H   CO   ,8H   HC   ,8H  NOX   ,8H   PT   ,
     . 8H  SO2   ,8H  POL6  /
      DATA ENPT,ENAR,ENLN,ABPT,ABAR,ABLN,ACPT,ACAR,ACLN / 12660*0.0 /
      DATA NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,NACAR,NACLN /9*0.0/
      END
```

FUNCTION CAVL

Purpose:

To compute the coupling coefficient at a receptor due to a line source.

Input:

Meteorological conditions: wind speed; stability; mixing height; critical distance for vertical mixing; psuedo downwind distances for horizontal and vertical spreads of the line source.

Source parameters: end point coordinates of the line (X-axis has been chosen to be along the wind vector); IAD flag for uniform or non-uniform line.

Receptor coordinates.

Output:

CAVL, the coupling coefficient.

Procedure:

1. Test whether the receptor is located with respect to the line source such that the concentration is negligible.
2. If the angle between the wind vector and line is sufficiently small, and the line is sufficiently long, set a flag for the line to be segmented.
3. Compute effective downwind distance and the horizontal and vertical dispersion coefficients.
4. Determine factor to be used in subdividing the line.
5. Test whether the line has a uniform density. If it is a runway used for aircraft arrival or departure (non-uniform density), call subroutine QMOD.
6. Determine the proper expression to be used and compute the concentration due to the line segment.
7. Test whether further segments need be considered. If not, output the concentration for the given receptor.

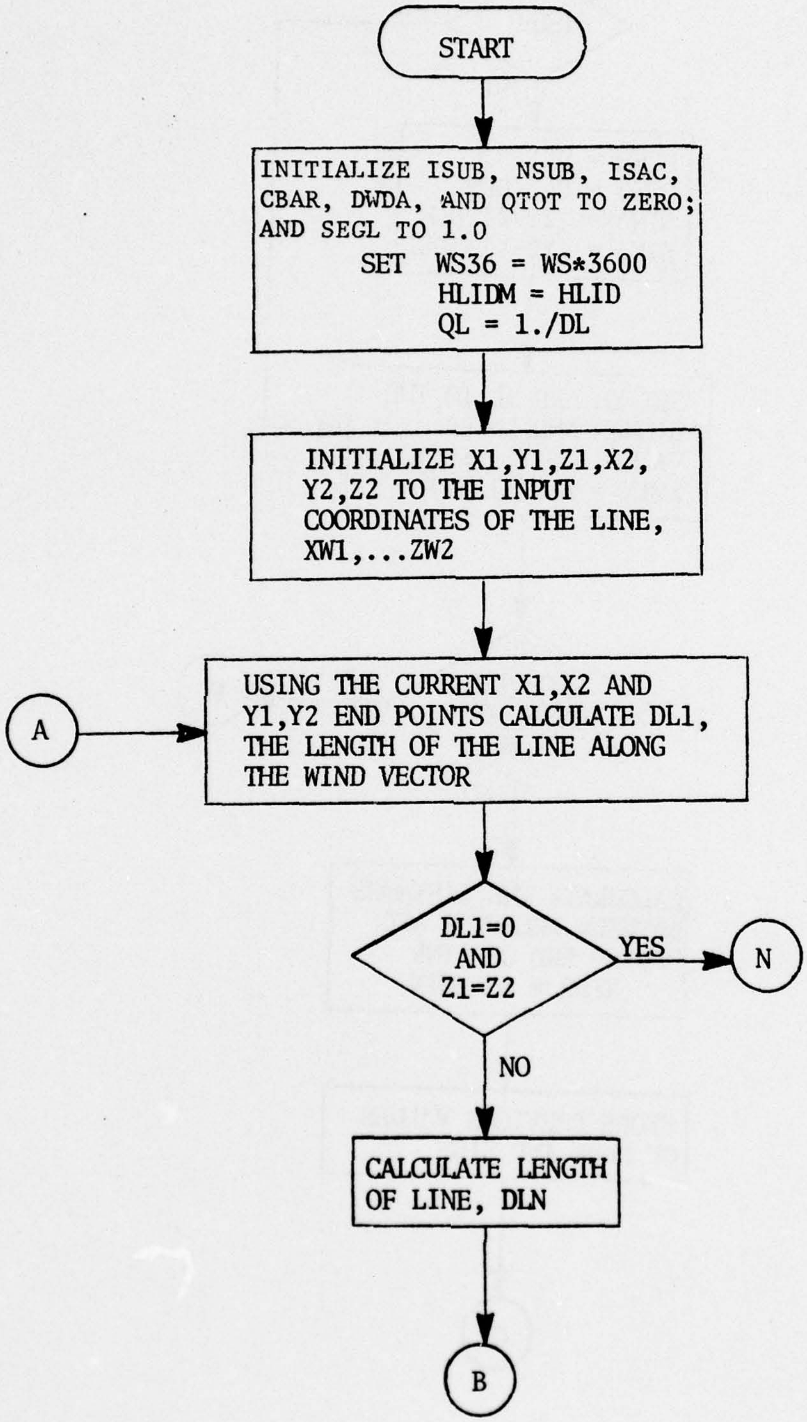
Functions Called:

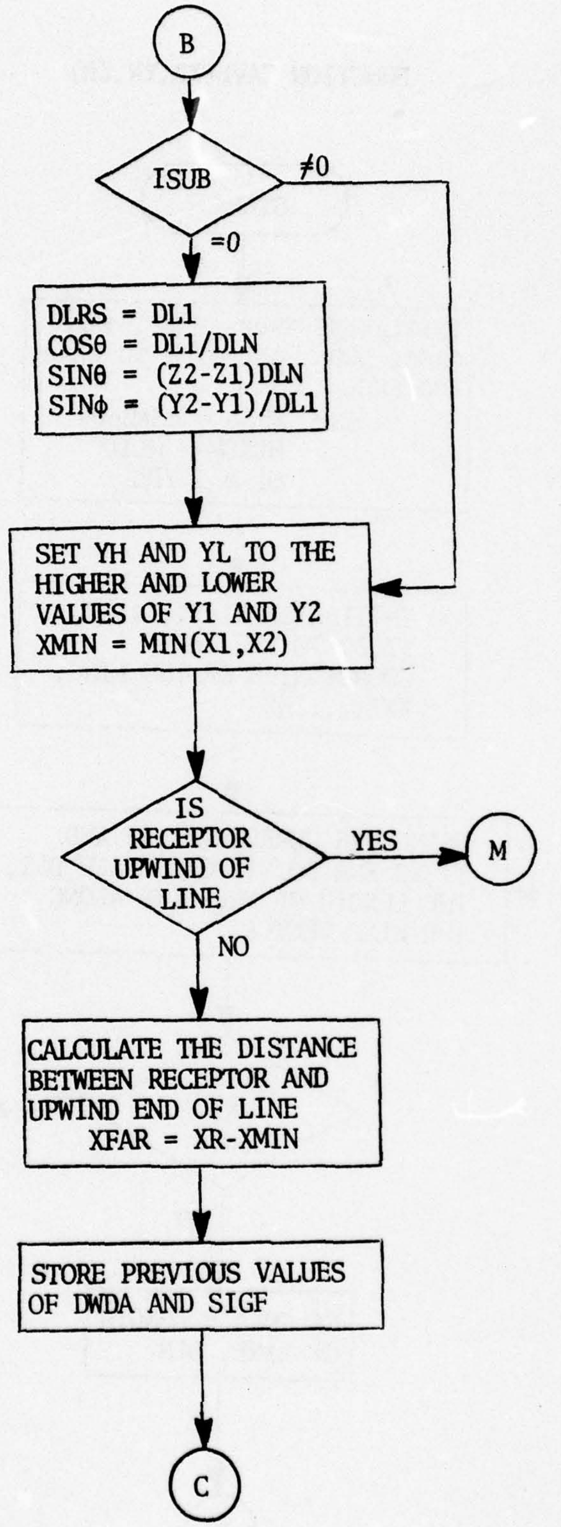
SIGY,SIGZ,DIFERF

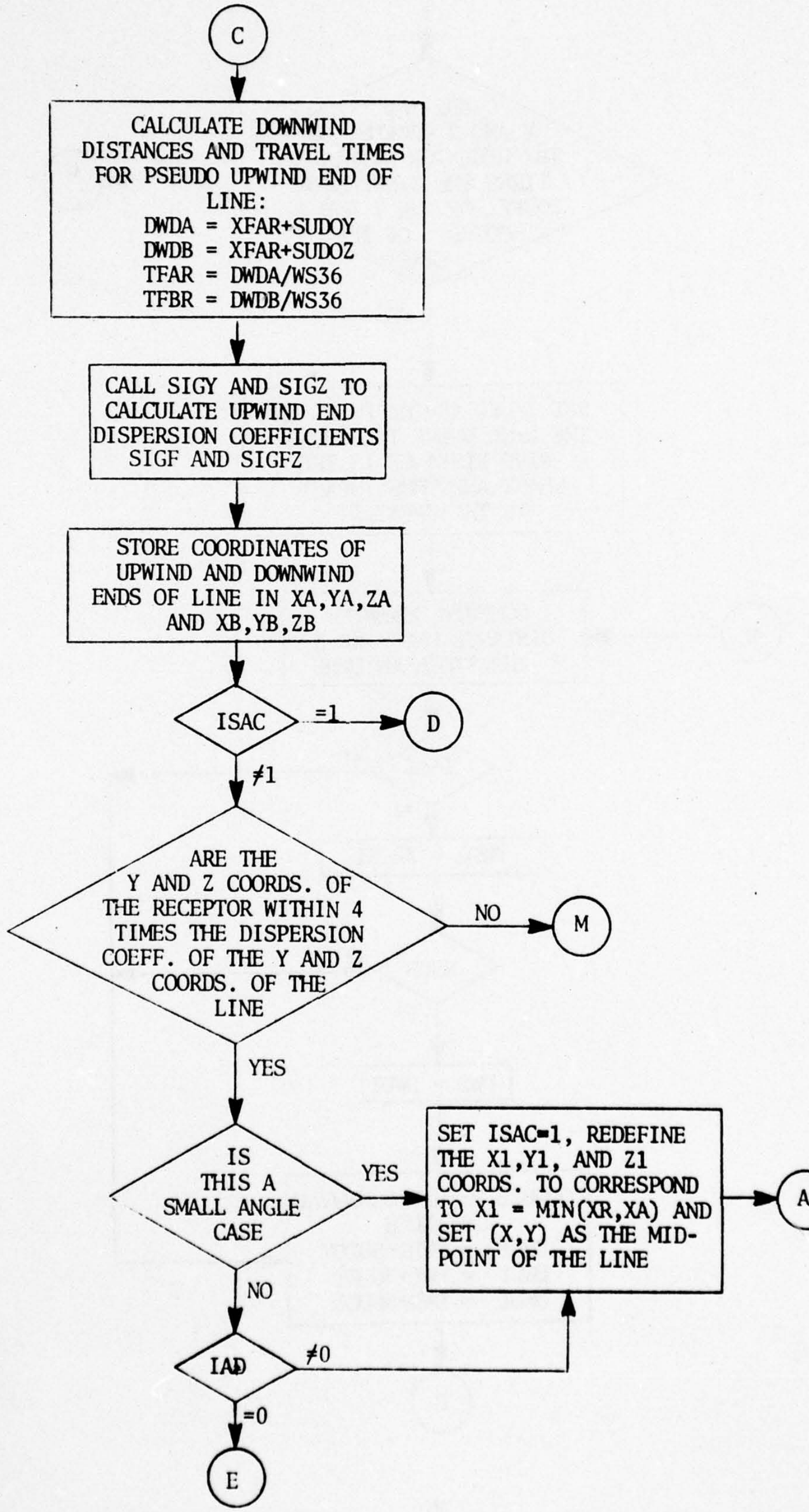
Subroutine Called:

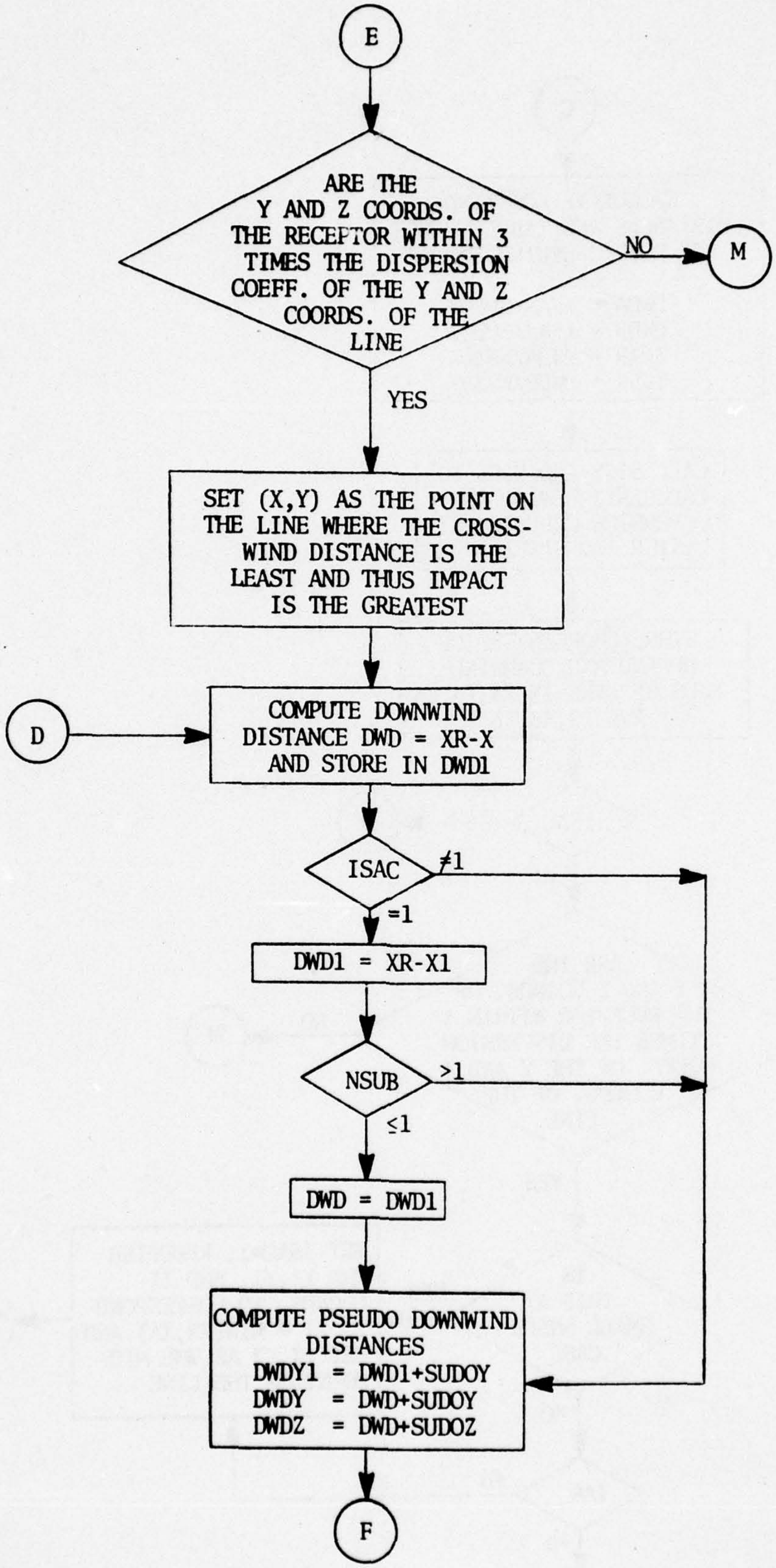
QMOD

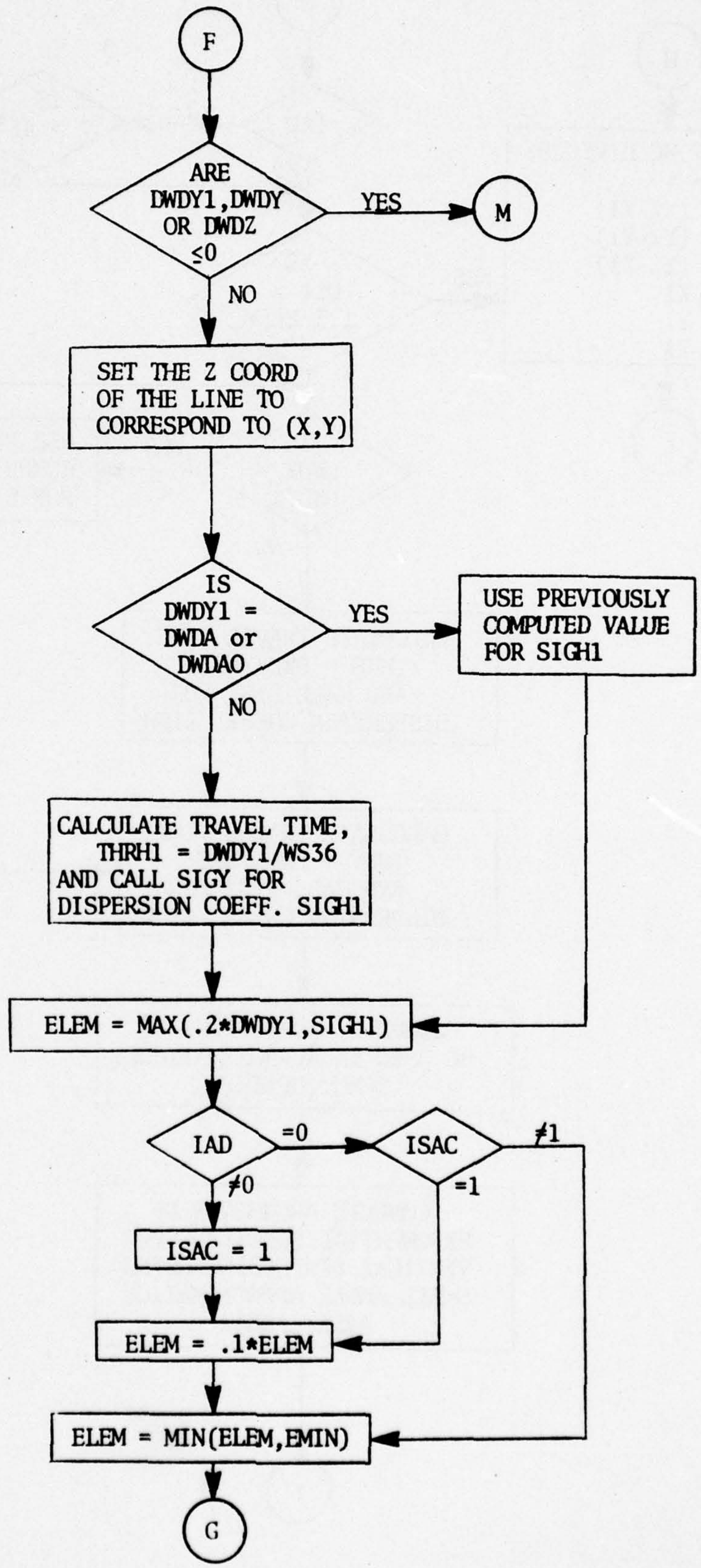
FUNCTION CAVL(XR,YR,ZR)

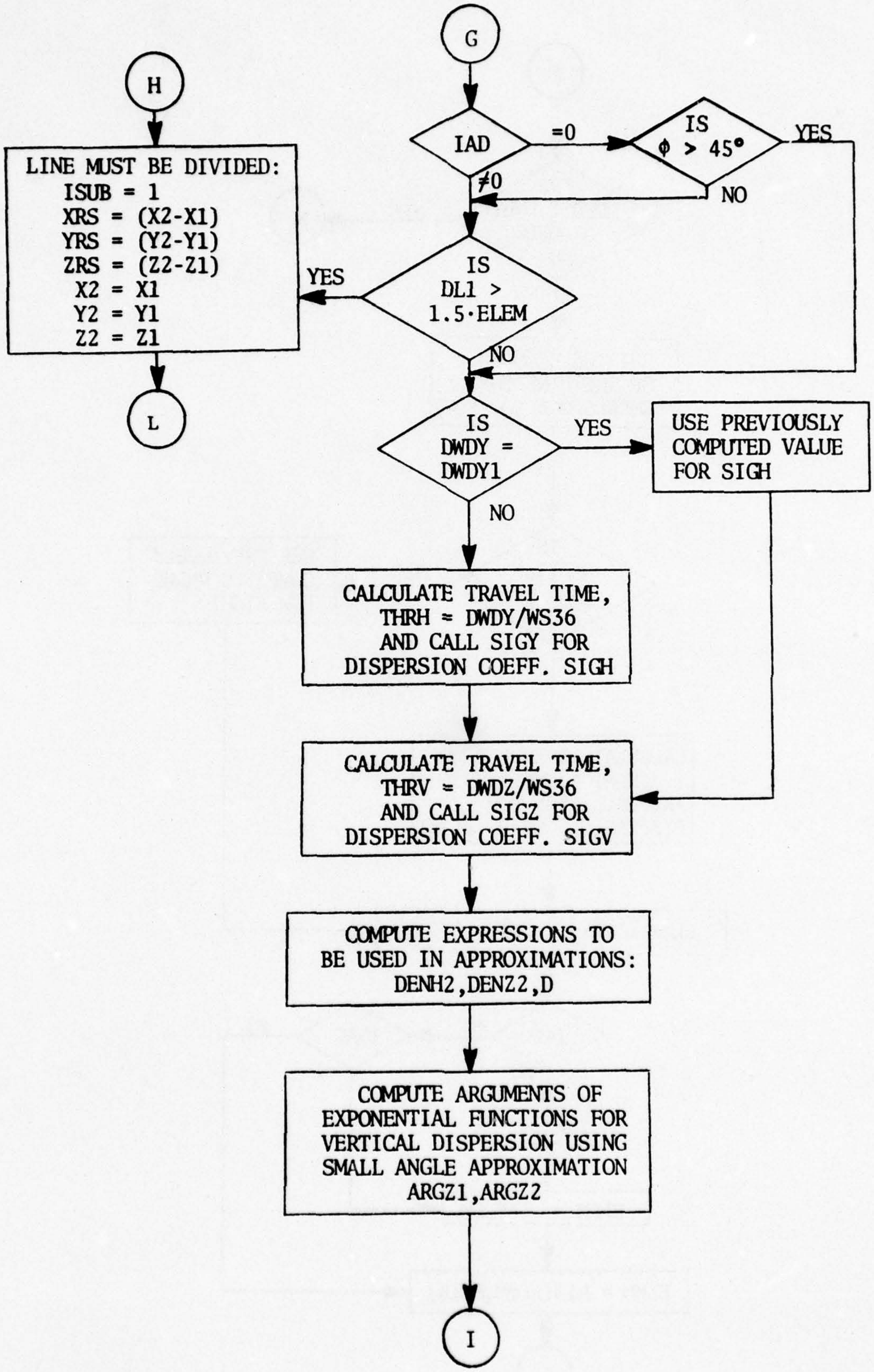


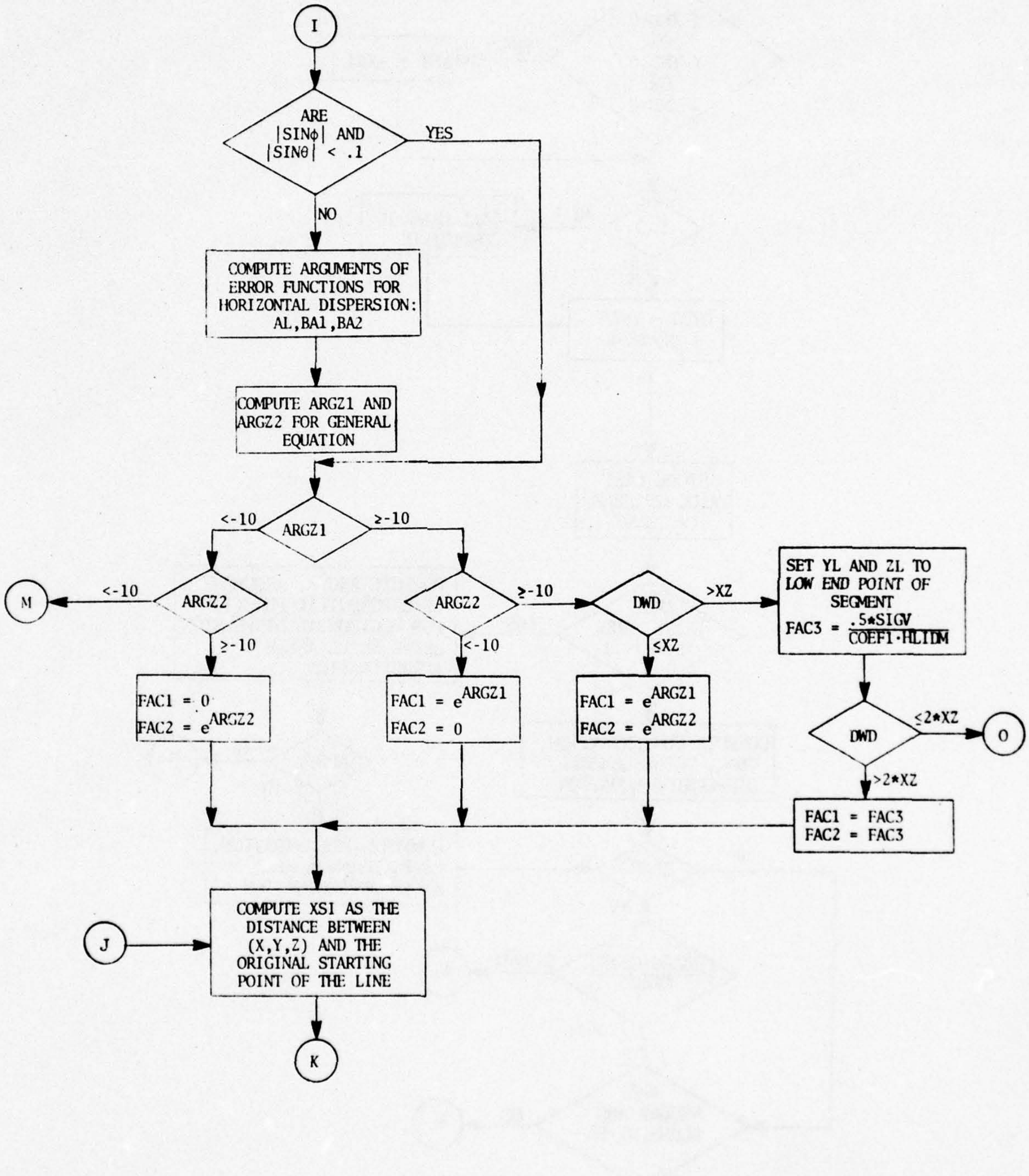

B
ISUB
≠0
=0
DLRS = DL1
COSθ = DL1/DLN
SINθ = (Z2-Z1)DLN
SINϕ = (Y2-Y1)/DL1
SET YH AND YL TO THE HIGHER AND LOWER VALUES OF Y1 AND Y2
XMIN = MIN(X1,X2)
IS RECEPTOR UPWIND OF LINE
YES
M
NO
CALCULATE THE DISTANCE BETWEEN RECEPTOR AND UPWIND END OF LINE
XFAR = XR-XMIN
STORE PREVIOUS VALUES OF DWDA AND SIGF
C

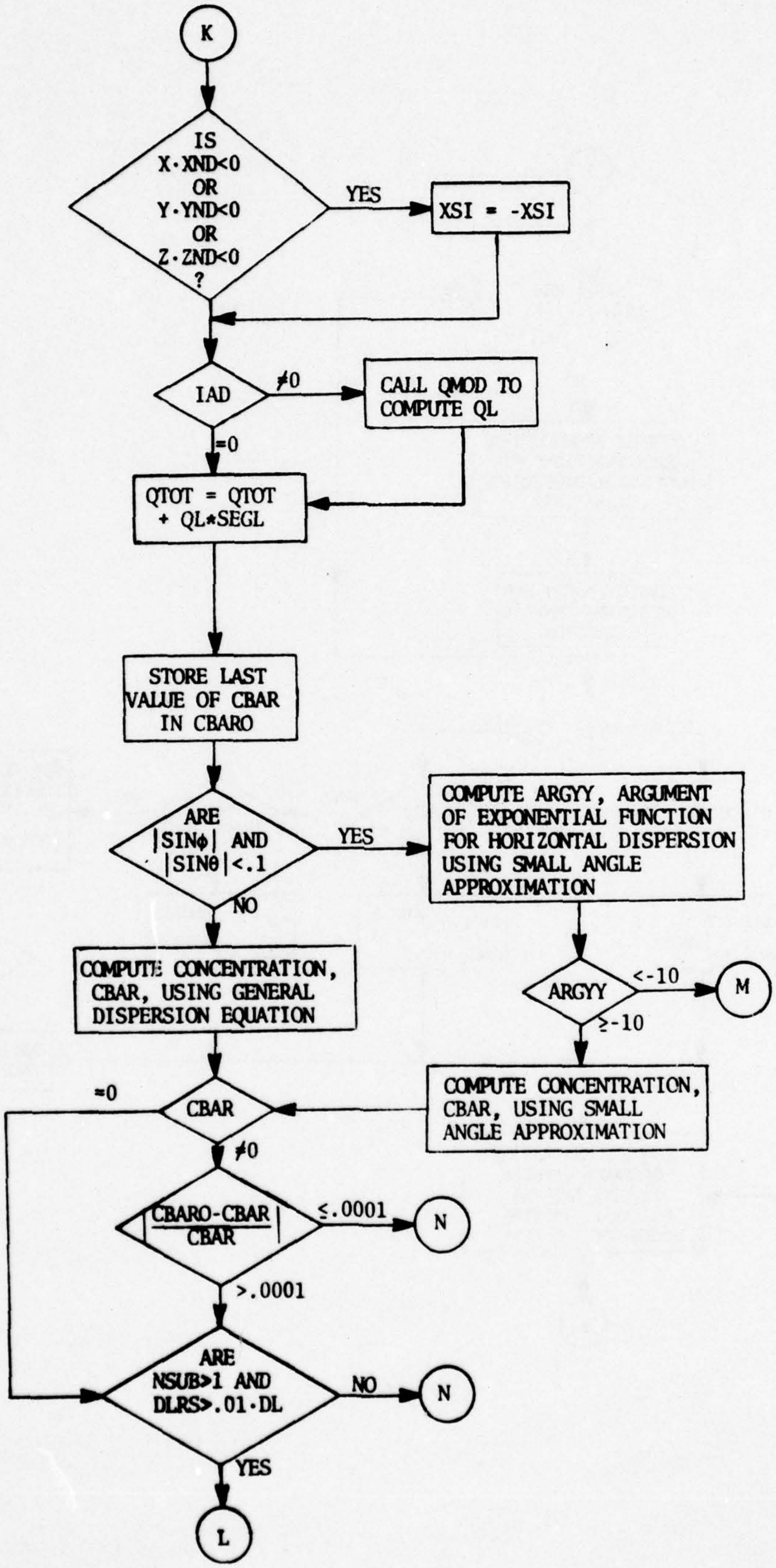

C
CALCULATE DOWNWIND DISTANCES AND TRAVEL TIMES FOR PSEUDO UPWIND END OF LINE:
DWDA = XFAR+SUDOY
DWDB = XFAR+SUDOZ
TFAR = DWDA/WS36
TFBR = DWDB/WS36
CALL SIGY AND SIGZ TO CALCULATE UPWIND END DISPERSION COEFFICIENTS SIGF AND SIGFZ
STORE COORDINATES OF UPWIND AND DOWNWIND ENDS OF LINE IN XA,YA,ZA AND XB,YB,ZB
ISAC
=1
D
≠1
ARE THE Y AND Z COORDS. OF THE RECEPTOR WITHIN 4 TIMES THE DISPERSION COEFF. OF THE Y AND Z COORDS. OF THE LINE
NO
M
YES
IS THIS A SMALL ANGLE CASE
YES
SET ISAC=1, REDEFINE THE X1,Y1, AND Z1 COORDS. TO CORRESPOND TO X1 = MIN(XR,XA) AND SET (X,Y) AS THE MID-POINT OF THE LINE
A
NO
IAD
≠0
=0
E


E
ARE THE Y AND Z COORDS. OF THE RECEPTOR WITHIN 3 TIMES THE DISPERSION COEFF. OF THE Y AND Z COORDS. OF THE LINE
NO
M
YES
SET (X,Y) AS THE POINT ON THE LINE WHERE THE CROSS-WIND DISTANCE IS THE LEAST AND THUS IMPACT IS THE GREATEST
D
COMPUTE DOWNWIND DISTANCE DWD = XR-X AND STORE IN DWD1
ISAC
≠1
=1
DWD1 = XR-X1
NSUB
>1
≤1
DWD = DWD1
COMPUTE PSEUDO DOWNWIND DISTANCES
DWDY1 = DWD1+SUDOY
DWDY = DWD+SUDOY
DWDZ = DWD+SUDOZ
F


F
ARE DWDY1,DWDY OR DWDZ ≤0
YES
M
NO
SET THE Z COORD OF THE LINE TO CORRESPOND TO (X,Y)
IS DWDY1 = DWDA or DWDAO
YES
USE PREVIOUSLY COMPUTED VALUE FOR SIGH1
NO
CALCULATE TRAVEL TIME, THRH1 = DWDY1/WS36 AND CALL SIGY FOR DISPERSION COEFF. SIGH1
ELEM = MAX(.2*DWDY1,SIGH1)
IAD
=0
ISAC
≠1
≠0
=1
ISAC = 1
ELEM = .1*ELEM
ELEM = MIN(ELEM,EMIN)
G


G
H
IAD
=0
IS
φ > 45°
YES
≠0
NO
LINE MUST BE DIVIDED:
ISUB = 1
XRS = (X2-X1)
YRS = (Y2-Y1)
ZRS = (Z2-Z1)
X2 = X1
Y2 = Y1
Z2 = Z1
YES
IS
DL1 >
1.5·ELEM
NO
L
IS
DWDY =
DWDY1
YES
USE PREVIOUSLY
COMPUTED VALUE
FOR SIGH
NO
CALCULATE TRAVEL TIME,
THRH = DWDY/WS36
AND CALL SIGY FOR
DISPERSION COEFF. SIGH
CALCULATE TRAVEL TIME,
THRV = DWDZ/WS36
AND CALL SIGZ FOR
DISPERSION COEFF. SIGV
COMPUTE EXPRESSIONS TO
BE USED IN APPROXIMATIONS:
DENH2,DENZ2,D
COMPUTE ARGUMENTS OF
EXPONENTIAL FUNCTIONS FOR
VERTICAL DISPERSION USING
SMALL ANGLE APPROXIMATION
ARGZ1,ARGZ2
I

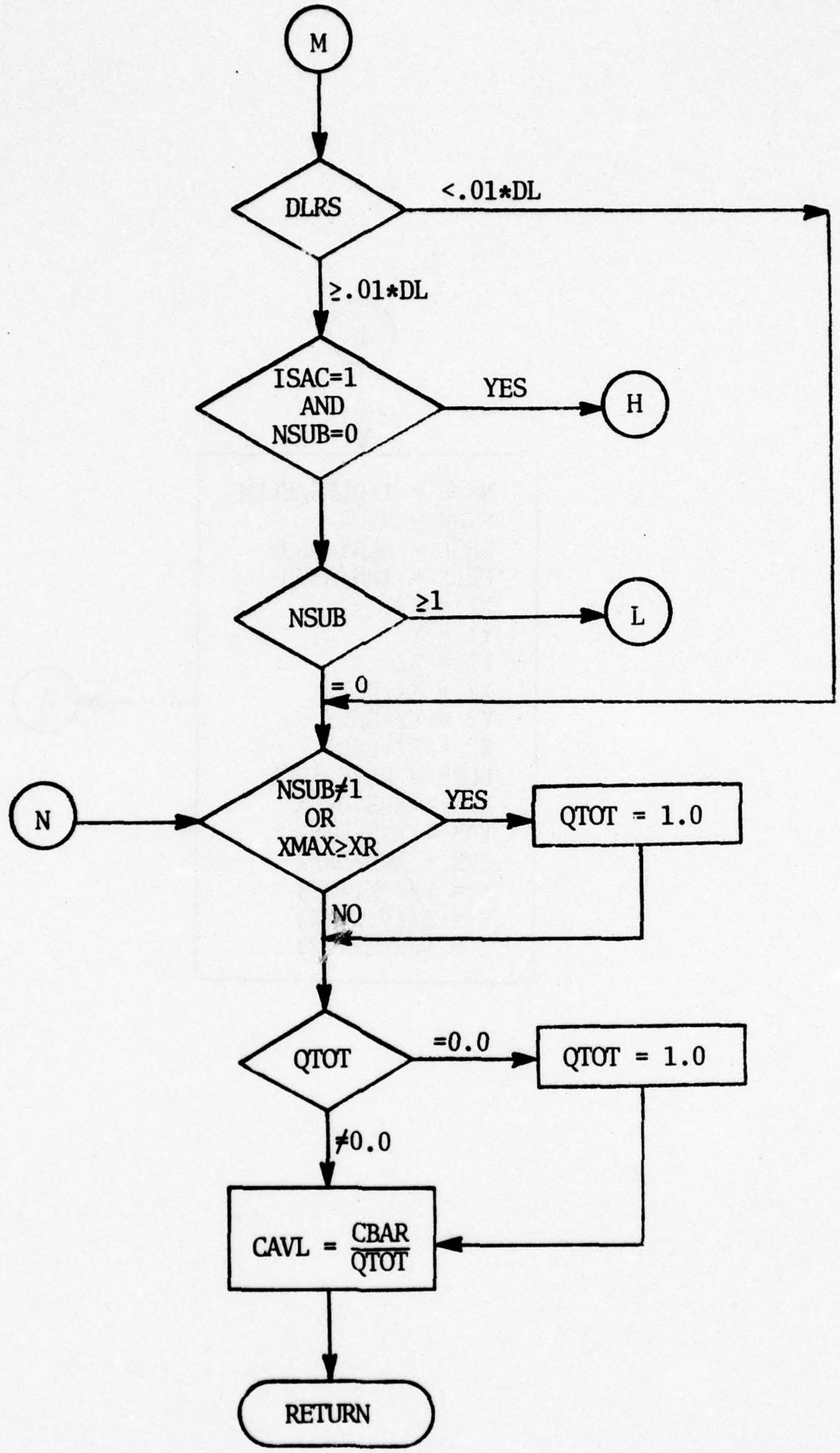

I
ARE |SINϕ| AND |SINθ| < .1
YES
NO
COMPUTE ARGUMENTS OF ERROR FUNCTIONS FOR HORIZONTAL DISPERSION: AL,BA1,BA2
COMPUTE ARGZ1 AND ARGZ2 FOR GENERAL EQUATION
ARGZ1
<-10
≥-10
ARGZ2
<-10
M
≥-10
FAC1 = 0
FAC2 = e^ARGZ2
ARGZ2
≥-10
<-10
FAC1 = e^ARGZ1
FAC2 = 0
DWD
>XZ
≤XZ
FAC1 = e^ARGZ1
FAC2 = e^ARGZ2
SET YL AND ZL TO LOW END POINT OF SEGMENT
FAC3 = .5*SIGV / COEF1·HLIDM
DWD
≤2*XZ
O
>2*XZ
FAC1 = FAC3
FAC2 = FAC3
J
COMPUTE XSI AS THE DISTANCE BETWEEN (X,Y,Z) AND THE ORIGINAL STARTING POINT OF THE LINE
K

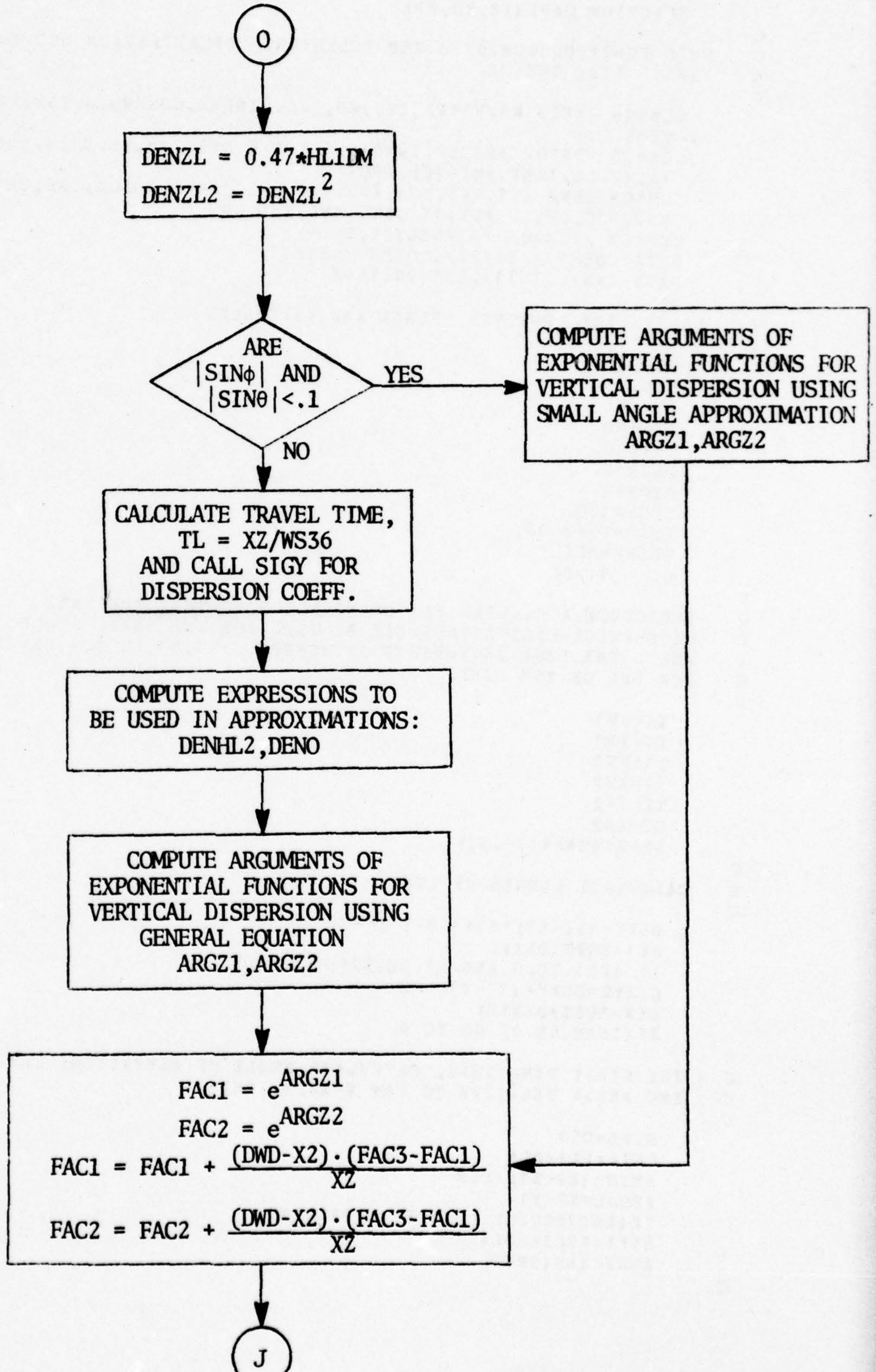

K
IS X·XND<0 OR Y·YND<0 OR Z·ZND<0 ?
YES
XSI = -XSI
IAD
≠0
CALL QMOD TO COMPUTE QL
=0
QTOT = QTOT + QL*SEGL
STORE LAST VALUE OF CBAR IN CBARO
ARE |SINφ| AND |SINθ|<.1
YES
COMPUTE ARGYY, ARGUMENT OF EXPONENTIAL FUNCTION FOR HORIZONTAL DISPERSION USING SMALL ANGLE APPROXIMATION
NO
COMPUTE CONCENTRATION, CBAR, USING GENERAL DISPERSION EQUATION
ARGYY
<-10
M
≥-10
COMPUTE CONCENTRATION, CBAR, USING SMALL ANGLE APPROXIMATION
=0
CBAR
≠0
|(CBARO-CBAR)/CBAR|
≤.0001
N
>.0001
ARE NSUB>1 AND DLRS>.01·DL
NO
N
YES
L

L

```
NSUB = 1+DLRS/ELEM
RSUB = NSUB
SEGL = DLRS/RSUB
DELX = XRS/RSUB
X1 = X2
Y1 = Y2
Z1 = Z2
X2 = X2+DELX
Y2 = Y2+DELY
Z2 = Z2+DELZ
DLRS = DLRS-SEGL
XRS = XRS-DELX
YRS = YRS-DELY
ZRS = ZRS-DELZ
X = 1/2(X1+X2)
Y = 1/2(Y1+Y2)
Z = 1/2(Z1+Z2)
```

A


M
DLRS
<.01*DL
≥.01*DL
ISAC=1
AND
NSUB=0
YES
H
NSUB
≥1
L
= 0
N
NSUB≠1
OR
XMAX≥XR
YES
QTOT = 1.0
NO
QTOT
=0.0
QTOT = 1.0
≠0.0
CAVL = CBAR / QTOT
RETURN


O
DENZL = 0.47*HL1DM
DENZL2 = DENZL^2
ARE
|SINφ| AND
|SINθ|<.1
YES
NO
COMPUTE ARGUMENTS OF EXPONENTIAL FUNCTIONS FOR VERTICAL DISPERSION USING SMALL ANGLE APPROXIMATION ARGZ1,ARGZ2
CALCULATE TRAVEL TIME, TL = XZ/WS36 AND CALL SIGY FOR DISPERSION COEFF.
COMPUTE EXPRESSIONS TO BE USED IN APPROXIMATIONS: DENHL2,DENO
COMPUTE ARGUMENTS OF EXPONENTIAL FUNCTIONS FOR VERTICAL DISPERSION USING GENERAL EQUATION ARGZ1,ARGZ2
$FAC1 = e^{ARGZ1}$
$FAC2 = e^{ARGZ2}$
$FAC1 = FAC1 + \frac{(DWD-X2)\cdot(FAC3-FAC1)}{XZ}$
$FAC2 = FAC2 + \frac{(DWD-X2)\cdot(FAC3-FAC1)}{XZ}$
J

```
      FUNCTION CAVL(XR,YR,ZR)
C
C     THIS FUNCTION COMPUTES THE POLLUTANT CONCENTRATION DUE TO A
C     FINITE LINE SOURCE
C
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X5,Y5,Z5,W,DELZ,X6,Y6,Z6,
     . V1,V2,DL,TIME,EMIS(6),NPOL
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDCY,SUDOZ,IAD,TAIL,B,V12,VS,
     . WS2,WSC,RR,SP,XST,YST,ZST,XND,YND,ZND
      COMMON /XTRAN/ XZ,WSMD,TY,TZ
      DATA COEF1 /.39894/,COEF2 /.31831/
      DATA CAN/0.7071/,EMIN/9.144/
C
C     INITIALIZE COUNTERS, FLAGS AND VARIABLES
C
      ISUB=0
      NSUB=0
      ISAC=0
      LSAC=0
      CBAR=0.
      DWDA=0.
      QTOT=0.
      SEGL=1.0
      WS36=WS*3600.
      HLIDM=HLID
      QL = 1./DL
C
C     INTRODUCE A GENERAL SET OF NOTATION SO THAT THE SAME
C     DISPERSION CALCULATION CAN BE USED FOR THE SMALL ANGLE CASE
C     WHERE THE LINE IS FURTHER SEGMENTED.  X1,Y1,Z1 NOW REFER TO THE
C     LOW END OF THE LINE.
C
      X1=XW1
      Y1=YW1
      Z1=ZW1
      X2=XW2
      Y2=YW2
      Z2=ZW2
      XMAX=AMAX1(X1,X2)
C
C     CALCULATE LENGTH OF LINE
C
    5 DLXY=(X2-X1)**2+(Y2-Y1)**2
      DL1=SQRT(DLXY)
      IF (DL1.EQ.0.AND.Z1.EQ.Z2) GO TO 600
      DLXYZ=DLXY+(Z2-Z1)**2
      DLN=SQRT(DLXYZ)
      IF(ISUB.NE.0) GO TO 6
C
C     THE FIRST TIME THRU, CALCULATE ANGLE OF ELEVATION, THETA,
C     AND ANGLE RELATIVE TO THE X-AXIS, PHI
C
      DLRS=DL1
      CSTH=DL1/DLN
      SNTH=(Z2-Z1)/DLN
      PROJL=Y2-Y1
      IF(ABS(PROJL).LT.1.E-20) PROJL=0.
      SNFI=PROJL/DL1
      ASNF=ABS(SNFI)
C
```

```
C     FIND HIGH AND LOW ENDS OF LINE AS PROJECTED ON THE X-Y PLANE
C
    6 CONTINUE
      IF(Y1.GT.Y2) GO TO 1
      YH=Y2
      YL=Y1
      GO TO 2
    1 YH=Y1
      YL=Y2
    2 CONTINUE
C
C     TEST THE RECEPTOR LOCATION RELATIVE TO THE LINE SOURCE
C
      XMIN=AMIN1(X1,X2)
      IF ((XMIN-XR).GE.0.5) GO TO 500
C
C     RECEPTOR IS DOWNWIND, FIND DISTANCE TO UPWIND END OF LINE
C
      XFAR=XR-XMIN
C
C     STORE PREVIOUS VALUES AND COMPUTE NEW DOWNWIND DISTANCES
C     AND TRAVEL TIMES FOR PSEUDO UPWIND END OF LINE
C
      DWDAO=DWDA
      SIGFO=SIGF
      DWDA=XFAR+SUDOY
      DWDB=XFAR+SUDOZ
      TFAR=DWDA/WS36
      TFBR=DWDB/WS36
C
C     COMPUTE UPWIND END DISPERSION COEFFICIENTS
C
      SIGF=SIGY(JSTAB,TFAR)
      SIGFZ=SIGZ(JSTAB,TFBR)
C
C     STORE LINE COORDINATES
C
      IF(X1.LE.X2) GO TO 21
      XA=X2
      YA=Y2
      XB=X1
      YB=Y1
      ZB=Z1
      GO TO 22
   21 XA=X1
      YA=Y1
      XB=X2
      YB=Y2
      ZB=Z2
   22 CONTINUE
      IF(ISAC.EQ.1) GO TO 4
C
C     ARE Y AND Z COORDS OF RECEPTOR WITHIN 4 TIMES THE DISPERSION
C     COEFFICIENT OF THE Y AND Z COORDS OF THE LINE
C
      IF(YR.GT.(YH+4.*SIGF)) GO TO 500
      IF(YR.LT.(YL-4.*SIGF)) GO TO 500
      IF(ZR.GT.(Z2+4.*SIGFZ)) GO TO 500
      IF(ZR.LT.(Z1-4.*SIGFZ)) GO TO 500
      IF (ASNF .LT.  CAN .AND. ABS(SNTH) .LT. CAN) GO TO 3
      IF (IAD.NE.0) GO TO 3
C
```

```
C     ANGLE IS LARGE:  ARE THE RECEPTOR COORDS WITHIN 3 TIMES THE
C     DISPERSION COEFFICIENT OF THE LINE COORDS.
C
        IF(YR.GT.(YH+3.*SIGF)) GO TO 500
        IF(YR.LT.(YL-3.*SIGF)) GO TO 500
        IF(ZR.GT.(Z2+3.*SIGFZ)) GO TO 500
        IF(ZR.LT.(Z1-3.*SIGFZ)) GO TO 500
C
C     SET (X,Y) AS POINT ON LINE WHERE IMPACT IS GREATEST
C
        X=X1+(YR-Y1)*(X2-X1)/(Y2-Y1)
        IF(X.GT.XB) GO TO 333
        IF(X.LT.XA) GO TO 33
        Y=YR
        GO TO 4
C
C     ANGLE IS SMALL:  REDEFINE LINE COORDS AND SET (X,Y) AS
C     MIDPOINT OF SEGMENT
C
      3 IF(ASNF.LT.0.1.AND.(ABS(SNTH)).LT.0.1) ISAC=1
        ISAC=1
     30 X=AMIN1(XR,XA)
        Y1=Y1+(X-X1)*(Y2-Y1)/(X2-X1)
        Z1=Z1+(X-X1)*(Z2-Z1)/(X2-X1)
        X1=X
        X2=XB
        Y2=YB
        Z2=ZB
        X=0.5*(X1+X2)
        Y=0.5*(Y1+Y2)
        GO TO 5
     33 X=XA
        Y=YA
        GO TO 4
    333 X=XB
        Y=YB
C
C     COMPUTE DOWNWIND DISTANCE
C
      4 DWD=XR-X
        IF (DWD.LT.-.01) GO TO 30
        DWD1=DWD
        IF (ISAC.NE.1) GO TO 40
        DWD1=XR-X1
        IF (NSUB.LE.1) DWD=DWD1
C
C     COMPUTE PSEUDO DOWNWIND DISTANCES
C
     40 DWDY1=DWD1+SUDOY
        DWDY=DWD+SUDOY
        DWDZ=DWD+SUDOZ
C
C     SET Z COORDINATE OF LINE
C
        IF(X1.EQ.X2) GO TO 44
        Z=Z1+(X-X1)*(Z2-Z1)/(X2-X1)
        GO TO 444
     44 Z=Z1+(Y-Y1)*(Z2-Z1)/(Y2-Y1)
    444 CONTINUE
C
C     COMPUTE TRAVEL TIME AND DISPERSION COEFFICIENT FOR
C     PSEUDO DOWNWIND DISTANCE
```

```
C
      IF (DWDY1.EQ.DWDA) GO TO 4111
      IF (DWDY1.EQ.DWDAO) GO TO 4113
      THRH1=DWDY1/WS36
      SIGH1=SIGY(JSTAB,THRH1)
 4211 CONTINUE
C
C    DETERMINE FACTOR TO BE USED IN SUB-DIVIDING THE LINE
C
      ELEM=AMAX1(0.2*DWDY1,SIGH1)
      IF (IAD.NE.0) ISAC=1
      IF(ISAC.EQ.1)ELEM=.1*ELEM
      IF(ELEM.LT.EMIN) ELEM=EMIN
C
C    BRANCH IF ANGLE IS SMALL AND LINE SOURCE IS LONG.
C
      IF (IAD.NE.0) GO TO 4311
      IF (ASNF.GE.CAN) GO TO 4312
 4311 IF (DL1.GT.(1.5*ELEM)) GO TO 55
C
C    COMPUTE TRAVEL TIME AND DISPERION COEFFICIENT FOR PSEUDO
C    VERTICAL DISTANCE
C
 4312 IF (DWDY.EQ.DWDY1) GO TO 4112
      THRH=DWDY/WS36
      SIGH=SIGY(JSTAB,THRH)
 4212 CONTINUE
      THRV=DWDZ/WS36
      SIGV=SIGZ(JSTAB,THRV)
C
C    EXPRESSIONS TO BE USED IN APPROXIMATIONS
C
      DENH2=2.*SIGH**2
      DENZ2=2.*SIGV**2
      D=SIGH*SIGV
C
C    ARGUMENTS OF EXPONENTIAL FUNCTION FOR VERTICAL DISPERSION
C    USING SMALL ANGLE APPPOXIMATION
C
      ARGZ1=-(ZR-Z1)**2/DENZ2
      ARGZ2=-(ZR+Z1)**2/DENZ2
      IF (LSAC.EQ.1) GO TO 446
      GO TO 445
 4111 SIGH1=SIGF
      GO TO 4211
 4112 SIGH=SIGH1
      GO TO 4212
 4113 SIGH1=SIGFO
      GO TO 4211
C
C    LARGE ANGLE CASE:  ARGUMENTS OF ERROR FUNCTIONS FOR
C    HORIZONTAL DISPERSION
C
  445 CONTINUE
      ARG=CSTH**2*SNFI**2*SIGV**2+SNTH**2*SIGH**2
      RARG=SQRT(ARG)
      A=RARG/(1.4142*D)
      AL=DLN*A
      ARG1=(YR-Y1)*CSTH*SNFI*SIGV**2
      ARG21=(ZR-Z1)*SNTH*SIGH**2
      ARG22=-(ZR+Z1)*SNTH*SIGH**2
      PA1=-(ARG1+ARG21)/(1.4142*D*RARG)
```

```
      BA2=-(ARG1+ARG22)/(1.4142*D*RARG)
      C1=(YR-Y1)**2/DENH2-ARGZ1
      C2=(YR-Y1)**2/DENH2-ARGZ2
C
C   ARGUMENTS OF EXPONENTIAL FUNCITONS FOR VERTICAL DISPERSION
C   USING THE GENERAL EQUATION
C
      ARGZ1=BA1**2-C1
      ARGZ2=BA2**2-C2
  446 IF(ARGZ1.LT.-10.)GOTO 2411
      IF(ARGZ2.GE.-10.)GOTO 2412
      FAC1=EXP(ARGZ1)
      FAC2=0
      GC TO 39
 2411 IF (ARGZ2.LT.-10) GO TO 500
      FAC1=0
      FAC2=EXP(ARGZ2)
      GC TO 39
 2412 IF(DWD.GT.XZ) GO TO 100
C
C   DOWNWIND DISTANCE IS LESS THAN THE CRITICAL DISTANCE:  ONLY
C   SOURCE AND GROUND REFLECTION ARE CONSIDERED
C
      FAC1=EXP(ARGZ1)
      FAC2=EXP(ARGZ2)
   39 CCNTINUE
C
C   FIND THE LINEAR DISTRIBUTION OF POLLUTION ON THE
C   RUNWAY FOR LANDING AND TAKE-OFF
C
      XSI2=(X-XST)**2+(Y-YST)**2+(Z-ZST)**2
      XSI=SQRT(XSI2)
      IF(X*XND.LT.0.OR.Y*YND.LT.0.OR.Z*ZND.LT.0) XSI=-XSI
      IF (IAD .NE. 0) CALL QMOD(XSI,QL)
      QTOT=QTOT+QL*SEGL
C
C   STORE LAST VALUE OF CBAR
C
      CBARO=CBAR
      IF(LSAC.EQ.1) GO TO 50
C
C   GENERAL DISPERSION EQUATION
C
      FJ1=FAC1*DIFERF(BA1,AL)
      FJ2=FAC2*DIFERF(BA2,AL)
      CBAR=CBAR+0.35355*COEF1*QI*(FJ1+FJ2)/(A*D)
  499 IF (CBARO.EQ.C) GO TO 49
      IF(ABS((CBARO-CBAR)/CBAR).LE..00010) GO TO 600
   49 CCNTINUE
      IF(NSUB.GT.1.AND.DLRS.GT.(.01*DL)) GO TO 60
      GC TO 600
C
C   SMALL-ANGLE APPROXIMATION
C
   50 ARGYY=-(YR-Y1)**2/DENH2
      IF(ARGYY.LT.-10.) GO TO 500
      FAC=0.5*(FAC1+FAC2)
      BRAC=EXP(ARGYY)
      CBAR=CBAR+COEF2*QL*DLN*FAC*BRAC/D
      GO TC 499
C
C   ANGLE IS SMALL AND SOURCE IS LONG
```

```
C
   55 ISUB=1
      XRS=X2-X1
      YRS=Y2-Y1
      ZRS=Z2-Z1
      X2=X1
      Y2=Y1
      Z2=Z1
C
C   COMPUTE COORDINATES FOR NEXT LINE SEGMENT
C
   60 NSUB=1.+DLRS/FLEM
      RSUB=NSUB
      SEGL=DLRS/RSUB
      DELX=XRS/RSUB
      DELY=YRS/RSUB
      DELZ=ZRS/RSUB
      X1=X2
      Y1=Y2
      Z1=Z2
      X2=X2+DELX
      Y2=Y2+DELY
      Z2=Z2+DELZ
      DLRS=DLRS-SEGL
      XRS=XRS-DELX
      YRS=YRS-DELY
      ZRS=ZRS-DELZ
      X=.5*(X1+X2)
      Y=.5*(Y1+Y2)
      Z=.5*(Z1+Z2)
C
C   GO BACK TO COMPUTE CONTRIBUTION FROM NEXT SEGMENT
C
      GO TO 5
C
C   DOWNWIND DISTANCE IS GREATER THAN, BUT LESS THAN TWICE, THE
C   CRITICAL DISTANCE.  LINEAR INTERPOLATION IS USED
C
  100 YL = Y1
      ZL = Z1
      IF (Z1 .LE. Z2) GO TO 105
      YL = Y2
      ZL = Z2
  105 FAC3=0.5*SIGV/(COEF1*HLIDM)
      IF(DWD.GT.2.*XZ) GO TO 200
      DENZL=0.47*HLIDM
      DENZL2=DENZL**2
      IF (ISAC.EQ.1) GO TO 101
  102 TL=XZ/WS36
      DENHL2=2.*SIGY(JSTAB,TL)**2
      DENO=CSTH**2*SNFI**2*DENZL2  +SNTH**2*DENHL2
      ARGZ1=-((YR-YL)*SNTH-(ZR-ZL)*CSTH*SNFI)**2/DENO
      ARGZ2=-((YR-YL)*SNTH-(ZR+ZL)*CSTH*SNFI)**2/DENO
      GO TO 103
  101 ARGZ1=-(ZR-ZL)**2/DENZL2
      ARGZ2=-(ZR+ZL)**2/DENZL2
  103 FAC1=EXP(ARGZ1)
      FAC2=EXP(ARGZ2)
      FAC1=FAC1+(DWD-XZ)*(FAC3-FAC1)/XZ
      FAC2=FAC2+(DWD-XZ)*(FAC3-FAC2)/XZ
      GO TO 39
C
```

```
C    DOWNWIND DISTANCE IS BEYOND 2 TIMES THE CRITICAL DISTANCE,
C    UNIFORM MIXING IS ASSUMED
C
  200 FAC1=FAC3
      FAC2=FAC3
      GO TO 39
  500 IF(DLRS.LT.(.01*DL)) GO TO 600
      IF(ISAC.EQ.1.AND.NSUB.EQ.0) GO TO 55
      IF(NSUB.GE.1) GO TO 60
  600 IF (NSUB.NE.1.OR.XMAX.GE.XP) QTOT=1.0
      IF (QTOT.EQ.0.0) QTOT=1.0
C
C    NORMALIZE CBAR TO THE TOTAL POLLUTANT DENSITY CALCULATED
C    ALONG THE LINE
C
      CAVL=CBAR/QTOT
      RETURN
      END
```

SUBROUTINE CLASSE

Purpose:

To print an error message if the wrong ICLASS value is input to one of the airbase non-aircraft or environ emission distribution subroutines.

Input:

None

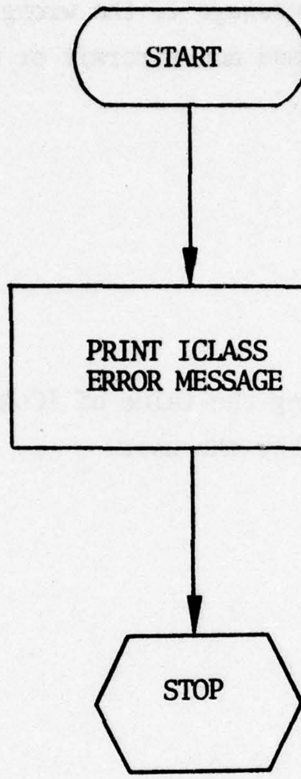
Output:

A message indicating the value of ICLASS set by the code and the value supplied by the user.

Subroutines Called:

None

SUBROUTINE CLASSE



```
      SUBFCUTINE CLASSE  (I,J)                                          CLASE000
C                                                                       CLASE001
C     THIS FOUTINE PRINTS THE ICLASS ERROR MESSAGE                      CLASE002
C                                                                       CLASE003
      PRINT 1, I,J                                                      CLASE004
    1 FORMAT(17H0ICLASS SHOULD BE,I4,18H, INPUT CARD READS,I4)          CLASE005
      STOP                                                              CLASE006
      END                                                               CLASE007
```

SUBROUTINE DEPART

Purpose:

To calculate the points in the runway roll and climbout modes as a function of aircraft type using current meteorological conditions and airbase specific pressure altitudes and airbase dependent basic aircraft parameters.

Input:

Basic aircraft data, current meteorological conditions, runway data, aircraft identification.

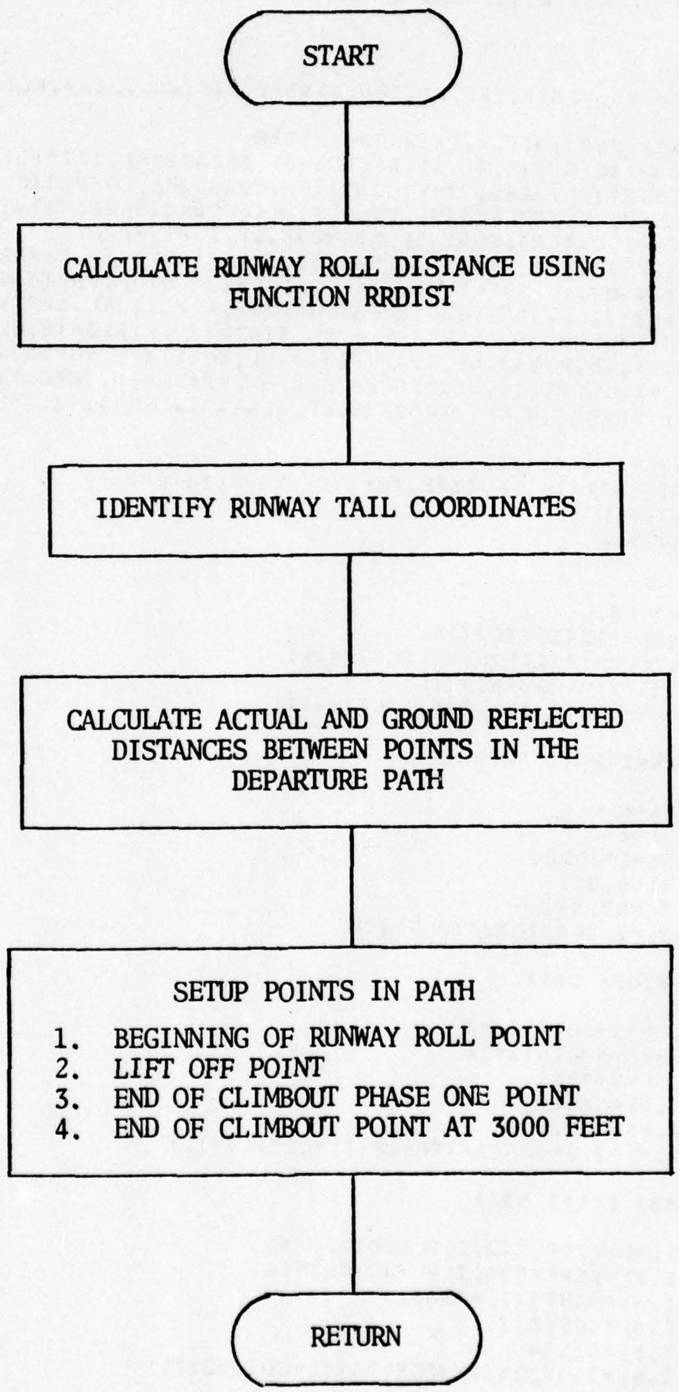
Output:

Points in departure path as a function of runway and aircraft type.

Subroutine Called:

Function RRDIST.

SUBROUTINE DEPART



```
      SUBROUTINE DEPART(N,I)
C
C     THIS ROUTINE CALCULATES THE POINTS IN THE DEPARTURE PATH
C     AS A FUNCTION OF RUNWAY(N) AND AIRCRAFT TYPE(I)
C
      REAL INDSPD
      INTEGER  ENGNO
      COMMON / MET / WS,WSMPH,IWS,WD,IWD,SINEWD,CVSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /ANNMET/ TBAR,ACD,P,PA,WSBAR,DTBAR
      COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),
     . INDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),
     . COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2),IDRR(8)
      COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),
     .ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),
     . IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),
     . IOBSEG(16,8,6),IDOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)
      RD=RNWY(7,N)
      WSPD=WS*1.9426*COS(WD-RD)
      HDIS12=RRDIST(IDRR(I),PA,TEMF,TOWT(I),WSPD)*3.048E-4
      XA=SIN(RNWY(7,N))
      YA=COS(RNWY(7,N))
      X=RNWY(2,N)
      Y=RNWY(3,N)
      Z=RNWY(4,N)/1000.
      DIS23=COHT1(I)/SIN(ASCNT1(I))
      DIS34=(CLMBHT-COHT1(I))/SIN(ASCNT2(I))
      HDIS23=COHT1(I)/TAN(ASCNT1(I))
      HDIS34=(CLMBHT-COHT1(I))/TAN(ASCNT2(I))
C
C     START OF RUNWAY ROLL DATA
C
      DEPFCN(1,I,N)=X
      DEPFCN(2,I,N)=Y
      DEPFCN(3,I,N)=Z*1000.
      DEPFCN(4,I,N)=0.0
      DEPFCN(5,I,N)=HDIS12
      DEPFCN(6,I,N)=2.0*HDIS12/TOSPD(I)
C
C     POINT OF LIFTOFF DATA
C
      DEPFCN(7,I,N)=X+HDIS12*XA
      DEPFCN(8,I,N)=Y+HDIS12*YA
      DEPFCN(9,I,N)=Z*1000.
      DEPFCN(10,I,N)=TOSPD(I)
      DEPFCN(11,I,N)=DIS23
      DEPFCN(12,I,N)=2.0*DIS23/(TOSPD(I)+COSPD1(I))
C
C     END OF CLIMB1 POINT DATA
C
      DEPFCN(13,I,N)=DEPFCN(7,I,N)+HDIS23*XA
      DEPFCN(14,I,N)=DEPFCN(8,I,N)+HDIS23*YA
      DEPFCN(15,I,N)=COHT1(I)*1000.
      DEPFCN(16,I,N)=COSPD1(I)
      DEPFCN(17,I,N)=DIS34
      DEPFCN(18,I,N)=2.0*DIS34/(COSPD1(I)+COSPD2(I))
C
C     END OF CLIMBOUT POINT DATA
```

```
C
      DEPFCN(19,I,N)=DEPFCN(13,I,N)+HDIS34*XA
      DEPFCN(20,I,N)=DEPFCN(14,I,N)+HDIS34*YA
      DEPFCN(21,I,N)=CLMBHT*1000.
      DEPFCN(22,I,N)=COSPD2(I)
      RETURN
      END
```

FUNCTION DIFERF(X,PH)

Purpose:

To find the difference between two error functions, erf(X+PH) - erf(X).

Input:

X and PH

Output:

The difference between the error functions

Procedure:

1. If PH ≤ .05, the formula given in the Handbook of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series 55 is used:

   $$DIFERF = 1.12838 \cdot PH \cdot e^{-X^2} [1 - PH \cdot X + (2 \cdot X^2 - 1) \cdot PH^{2/3}]$$

2. If PH > .05 and X and X+PH are of different sign:

   DIFERF = erf(X+PH) - erf(X)

3. If PH > .05 and X and X+PH are both negative:

   DIFERF = -1.*[erfc(-X) - erfc(-X-PH)]

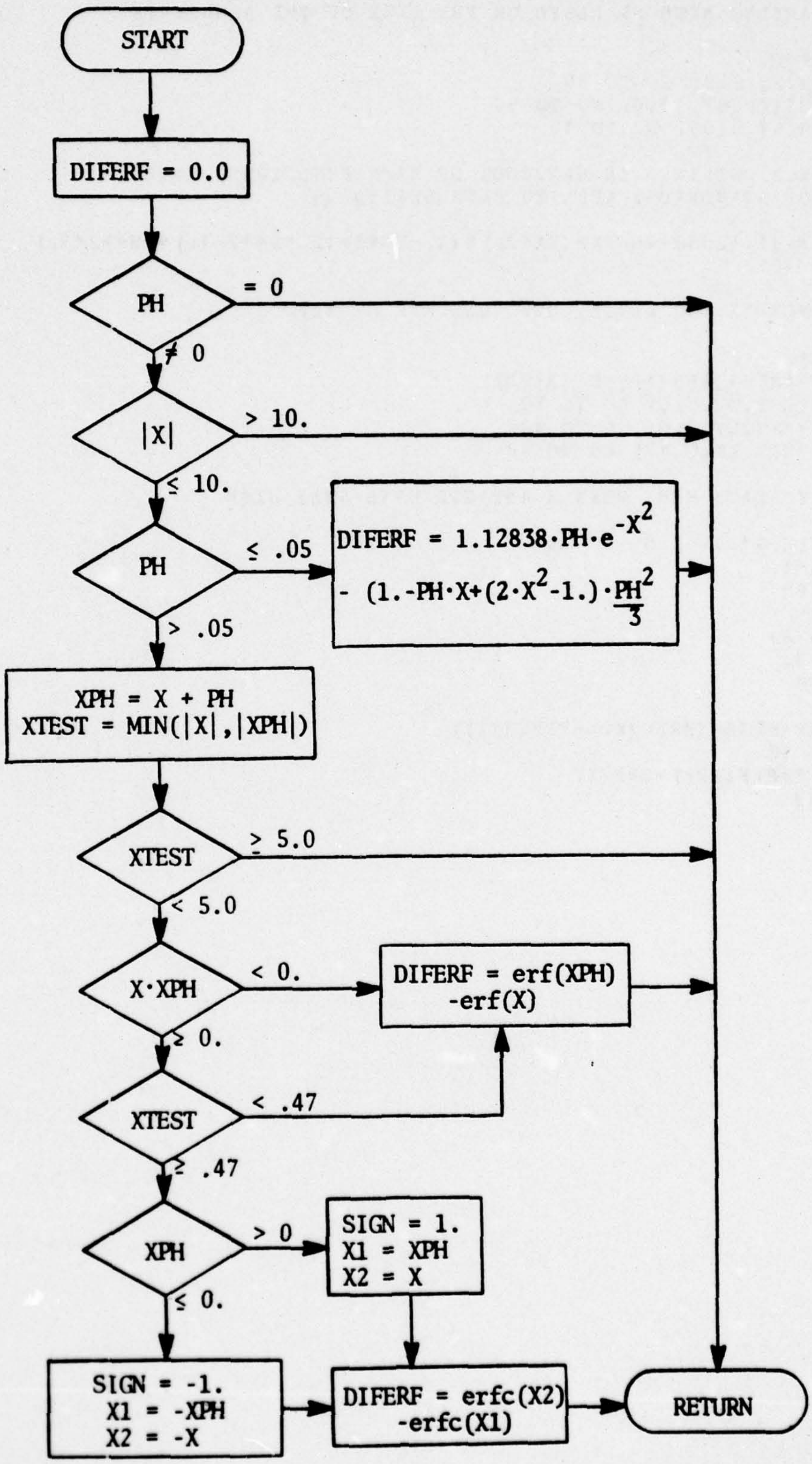
4. If PH > .05 and X and X+PH are both positive:

   DIFERF = erfc(X) - erfc(X+PH)

Function Called:

ERF, ERFC

FUNCTION DIFERF(X,PH)



```
      FUNCTION DIFERF(X,PH)
C
C    THIS FUNCTION FINDS THE DIFFERENCE BETWEEN TWO ERROR FUNCTIONS
C    USING VARYING METHODS BASED ON THE SIZE OF THE ARGUMENTS
C
      DIFERF=0.
      IF (PH.EQ.0.0) GO TO 50
      IF (ABS(X).GT.10.0) GO TO 50
      IF (PH.GT.0.05) GO TO 10
C
C    USE METHOD OUTLINED IN HANDBOOK OF MATH FUNCTIONS, NATL
C    BUREAU OF STANDARDS, APPLIED MATH SERIES 55
C
      DIFERF=(1.12838*PH/EXP(X**2))*(1.-PH*X+(2.*X**2-1.)*PH**2/3.)
      GO TO 50
C
C    DIFFERENCE IS TOO LARGE, MUST USE ERF OR ERFC
C
   10 XPH=X+PH
      XTEST=AMIN1(ABS(X),ABS(X+PH))
      IF (XTEST.GE.5.0) GO TO 50
      IF (X*XPH.LT.0.0) GO TO 40
      IF (XTEST.LT.0.47) GO TO 40
C
C    CAN ONLY REACH HERE WHEN X AND XPH HAVE SAME SIGN
C
      IF (XPH.GT.0.0) GO TO 20
      SIGN=-1.
      X1=-XPH
      X2=-X
      GO TO 30
   20 SIGN=1.
      X1=XPH
      X2=X
   30 DIFERF=SIGN*(ERFC(X2)-ERFC(X1))
      GO TO 50
   40 DIFERF=ERF(XPH)-ERF(X)
   50 RETURN
      END
```

SUBROUTINE EMISAR

Purpose:

To add emissions from a given activity to all others contained in the specified geometric area or line.

Input:

None

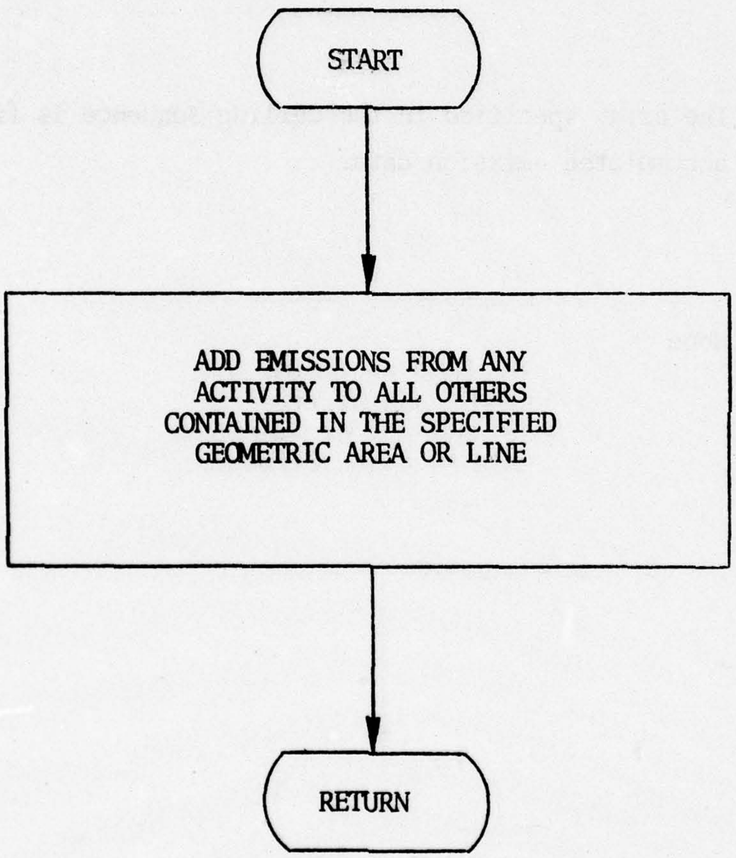
Output:

The array specified in the calling sequence is filled with the accumulated emission data.

Subroutines Called:

None

SUBROUTINE EMISAR



```
      SUBROUTINE EMISAR(MAXN,ARRAY,I1,I2)
C
C     THIS ROUTINE ACCUMULATES EMISSIONS FROM ANY ACTIVITY WITH
C     OTHERS CONTAINED IN THE SAME AIRBASE AREA OR LINE.
C        MAXN        = NO. OF SOURCES IN AN ACTIVITY
C        ARRAY       = SPECIFIED AREA OR LINE OUTPUT ARRAY
C        I1,I2       = DIMENSIONS OF ARRAY
C        NSRCE       = POINTER TO LOCATION OF SOURCES IN SORCE
C        LOC1        = POINTER TO LOCATION OF LIST OF EMISSIONS IN ARRAY
C        SORCE(2,N)  = POINTER TO LOCATION OF SOURCE AREA OR LINE
C
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      DIMENSION ARRAY(I1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
C
      DO 10 N=LSRCE,NSRCE
      J=SORCE(2,N)
      DO 10 I=1,NPLTS
      ARRAY(I+LOC1,J)=ARRAY(I+LOC1,J)+SORCE(I+2,N)
   10 CONTINUE
      RETURN
      END
```

SUBROUTINE ENARAY

Purpose:

1. To read from the master source tape all data needed to define environ point, area and line sources.

2. To compute the emission rates due to point sources, stationary, mobile, land use or combined area sources and roadway and non-roadway line sources.

Input:

If the diurnal distribution cards are input, an additional parameter, IMETH, is input here to choose the method of distribution of emissions from those land use or combined area source activities not using the default of a uniform distribution.

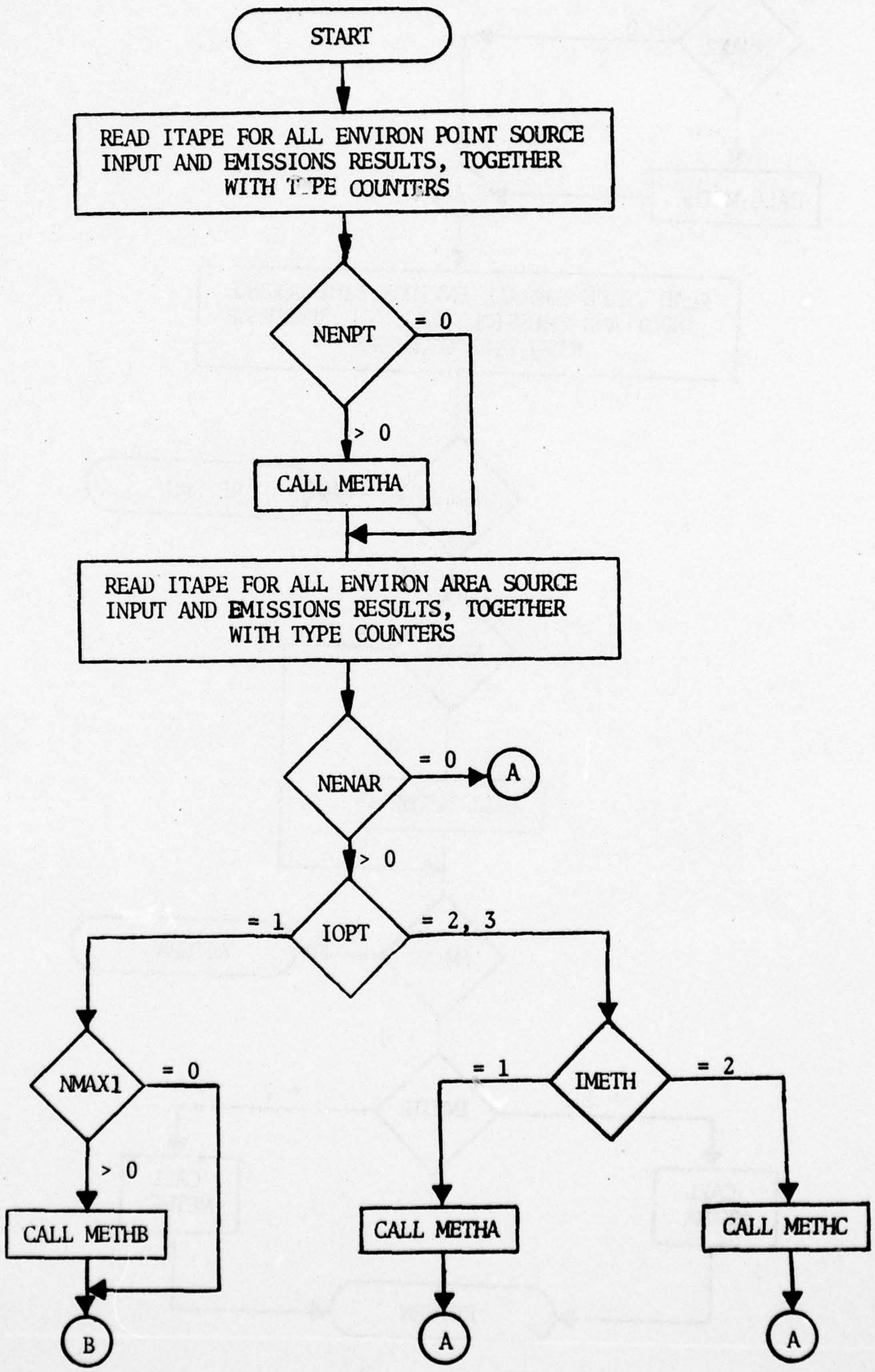
Output:

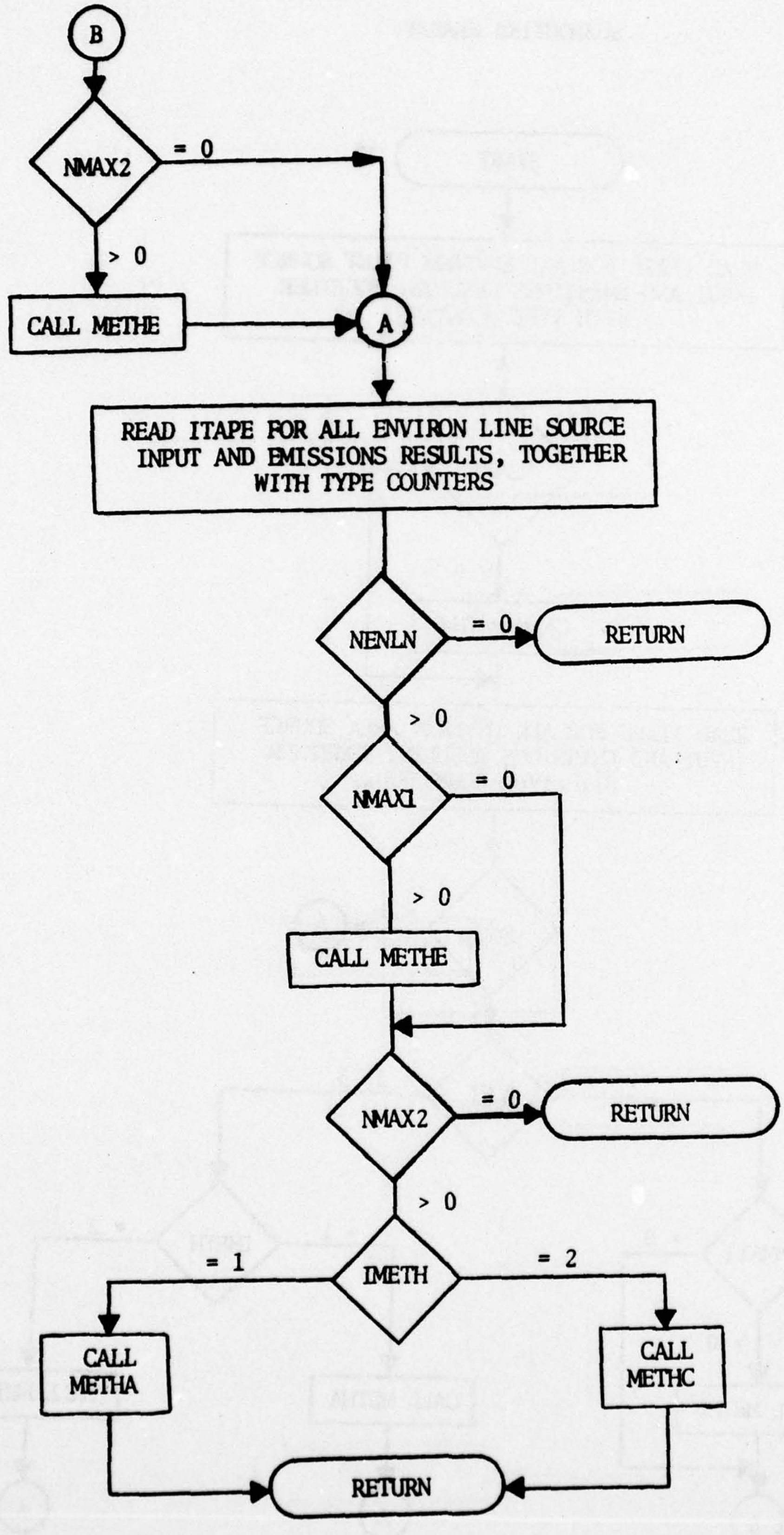
The arrays, ENPT, ENAR, and ENLN, are filled with geometry and emission data for all environ sources.

Subroutines Called:

METHA, METHB, METHC, METHE

SUBROUTINE ENARAY





B
NMAX2
= 0
> 0
CALL METHE
A
READ ITAPE FOR ALL ENVIRON LINE SOURCE INPUT AND EMISSIONS RESULTS, TOGETHER WITH TYPE COUNTERS
NENLN
= 0
RETURN
> 0
NMAX1
= 0
> 0
CALL METHE
NMAX2
= 0
RETURN
> 0
IMETH
= 1
= 2
CALL METHA
CALL METHC
RETURN

```
      SUBROUTINE ENARAY
C
C     THIS ROUTINE COMPUTES THE EMISSSION RATES FOR ALL
C     ENVIRON SOURCES
C
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON/MONMET/TMBAR,WSMBAR,AMDMBR,DTMBAR
      COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,
     .  JSTAB,HLID,TEMP,TEMK
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                   ABPT(16,150),ABAR(11,100),ABLN(14,100)
C
C****POINTS
C
      READ(ITAPE) NENPT,NTOT,((SORCE(I,N),I=1,NTOT),N=1,NENPT)
      IF (NENPT.EQ.0) GO TO 100
      ICLASS=201
      LOC1=10
      LOC2=11
      NGEOM=9
      I1=16
      I2=100
      IPT=1
      NSRCE=0
      CALL METHA(NENPT,ENPT,I1,I2,ICLASS)
C
C****AREAS
C
  100 READ(ITAPE) NENAR,NTOT,IOPT,NMAX1,NMAX2,
     1  ((SORCE(I,N),I=1,NTOT),N=1,NENAR)
      IF (NENAR.EQ.0) GO TO 300
      LOC1=5
      LOC2=7
      NGEOM=5
      IPT=0
      I1=11
      I2=100
      NSRCE=0
      GO TO (110,120,130),IOPT
C
C****OPTION 1  STATIONARY AREAS
C
  110 ICLASS=202
      IF (NMAX1.GT.0)
     1 CALL METHB(NMAX1,ENAR,I1,I2,ICLASS)
C
C****OPTION 1  MOBILE AREAS
C
      IF (NMAX2.GT.0)
     1 CALL METHE(NMAX2,ENAR,CVENMO,CVENDY,CVENHR,I1,I2)
      GO TO 300
C
C****OPTION 2 OR 3  LAND USE OR COMBINED AREAS
C
  120 ICLASS=203
```

```
      GO TO 200                                                         ENARY062
  130 ICLASS=204                                                        ENARY063
  200 IMETH=1                                                           ENARY064
      IF (JFLAG.EQ.0) READ 201,IMETH                                    ENARY065
  201 FORMAT(I4)                                                        ENARY066
      IF (IMETH.EQ.1) CALL METHA(NMAX1,ENAR,I1,I2,ICLASS)               ENARY067
      IF (IMETH.EQ.2) CALL METHC(NMAX1,ENAR,I1,I2,ICLASS)               ENARY068
C                                                                       ENARY069
C****LINES                                                              ENARY070
C       NMAX1 = NO. OF  ROADWAY LINES                                   ENARY071
C       NMAX2 = NO. OF NON-ROADWAY LINES                                ENARY072
C                                                                       ENARY073
  300 READ(ITAPE) NENLN,NTOT,NMAX1,NMAX2,                               ENARY074
     1  ((SORCE(I,N),I=1,NTOT),N=1,NENLN)                               ENARY075
      IF (NENLN.EQ.0) GO TO 400                                         ENARY076
      LCC1=8                                                            ENARY077
      LCC2=10                                                           ENARY078
      NGEOM=8                                                           ENARY079
      NSRCE=0                                                           ENARY080
      I1=14                                                             ENARY081
      I2=20                                                             ENARY082
      IPT=0                                                             ENARY083
      IF (NMAX1.GT.0)                                                   ENARY084
     1 CALL METHE(NMAX1,ENLN,CVENMO,CVENDY,CVENHR,I1,I2)                ENARY085
C                                                                       ENARY086
      IF (NMAX2.EQ.0) GO TO 400                                         ENARY087
      ICLASS=206                                                        ENARY088
      IMETH=1                                                           ENARY089
      IF (JFLAG.EQ.0) READ 201,IMETH                                    ENARY090
      IF (IMETH.EQ.1) CALL METHA(NMAX2,ENLN,I1,I2,ICLASS)               ENARY091
      IF (IMETH.EQ.2) CALL METHC(NMAX2,ENLN,I1,I2,ICLASS)               ENARY092
  400 RETURN                                                            ENARY093
      END                                                               ENARY094
```

SUBROUTINE INDINP
ENTRY DEPINP

Purpose:

To print the input parameters for both wind independent and wind dependent sources.

Input:

All source parameters.

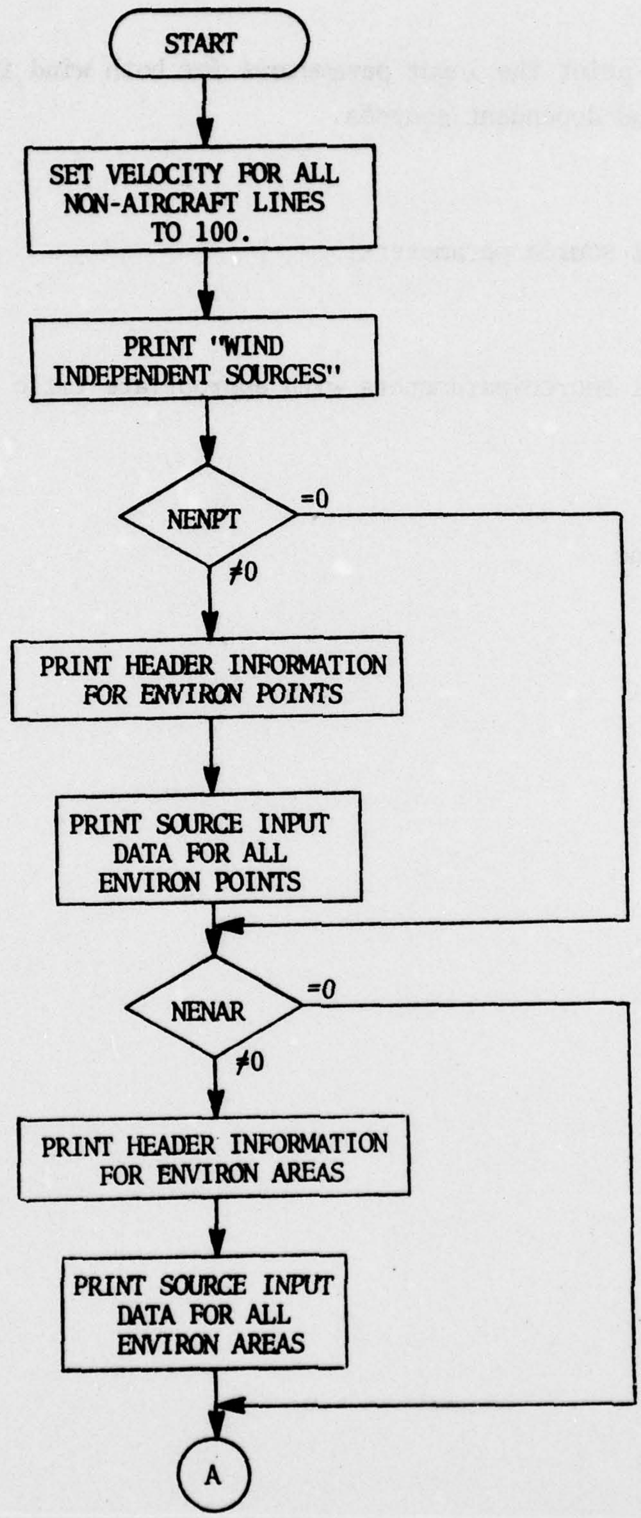
Output:

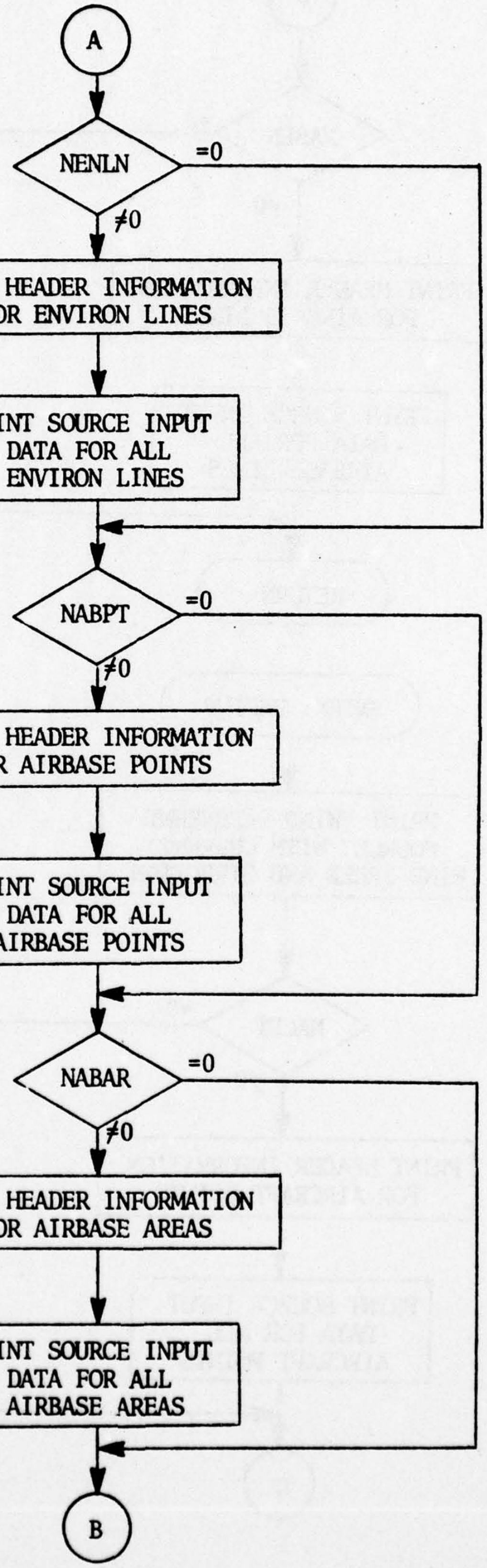
All source parameters with appropriate title information.

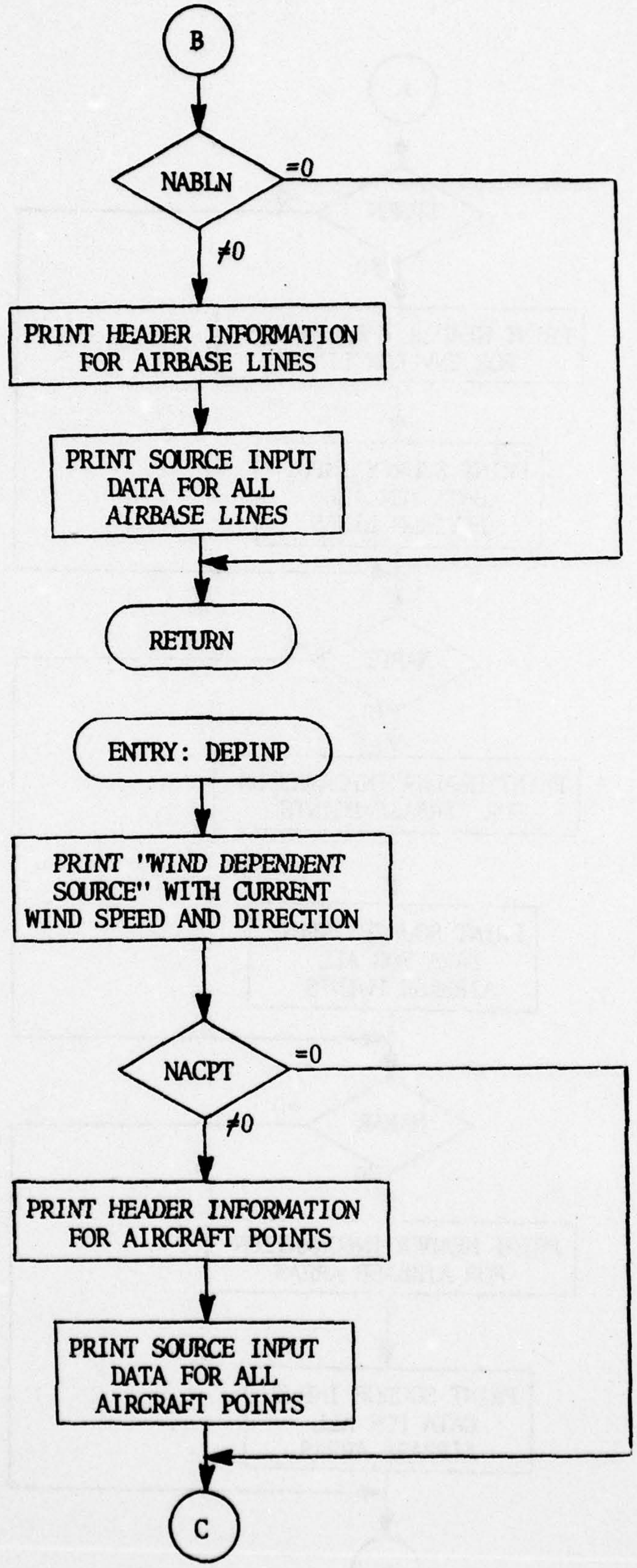
Subroutines Called:

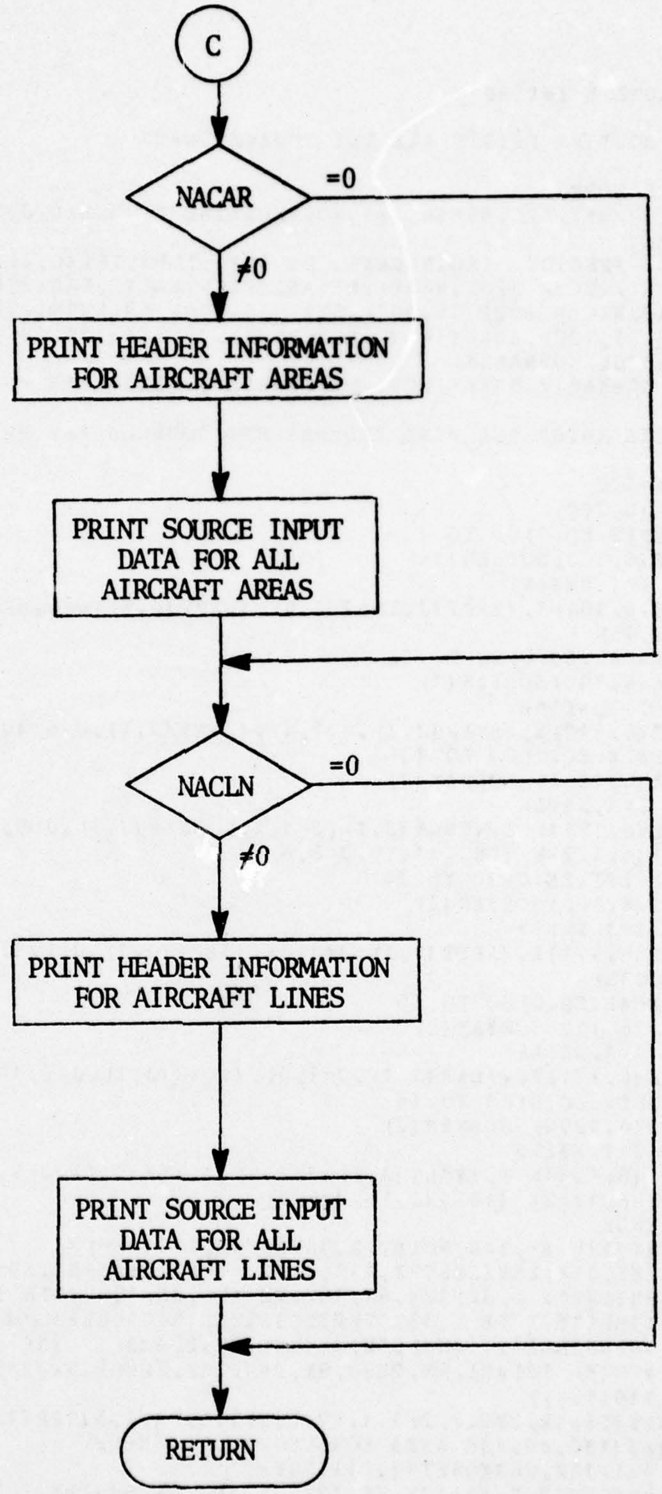
None

SUBROUTINE INDINP
(ENTRY: DEPINP)




A
NENLN
=0
≠0
PRINT HEADER INFORMATION FOR ENVIRON LINES
PRINT SOURCE INPUT DATA FOR ALL ENVIRON LINES
NABPT
=0
≠0
PRINT HEADER INFORMATION FOR AIRBASE POINTS
PRINT SOURCE INPUT DATA FOR ALL AIRBASE POINTS
NABAR
=0
≠0
PRINT HEADER INFORMATION FOR AIRBASE AREAS
PRINT SOURCE INPUT DATA FOR ALL AIRBASE AREAS
B


B
NABLN
=0
≠0
PRINT HEADER INFORMATION FOR AIRBASE LINES
PRINT SOURCE INPUT DATA FOR ALL AIRBASE LINES
RETURN
ENTRY: DEPINP
PRINT "WIND DEPENDENT SOURCE" WITH CURRENT WIND SPEED AND DIRECTION
NACPT
=0
≠0
PRINT HEADER INFORMATION FOR AIRCRAFT POINTS
PRINT SOURCE INPUT DATA FOR ALL AIRCRAFT POINTS
C


C
NACAR
=0
≠0
PRINT HEADER INFORMATION FOR AIRCRAFT AREAS
PRINT SOURCE INPUT DATA FOR ALL AIRCRAFT AREAS
NACLN
=0
≠0
PRINT HEADER INFORMATION FOR AIRCRAFT LINES
PRINT SOURCE INPUT DATA FOR ALL AIRCRAFT LINES
RETURN

```
      SUBROUTINE INDINP
C
C     THIS ROUTINE PRINTS ALL THE SOURCE INPUT
C
      REAL*8 SORNAM
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF
     . ,TEMK
      COMMON /PERIOD/ IMO,NCDAYS,IDY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      DIMENSION SORNAM(3)
      DATA SORNAM / 8HENVIRON ,8HAIRBASE ,8HAIRCRAFT /
C
C     AT THIS ENTRY ALL WIND INDEPENDENT SOURCES ARE PRINTED
C
      ENLNA=1.0
      WRITE(6,200)
      IF(NENPT.EQ.0)GO TO 11
      WRITE(6,100)SORNAM(1)
      DO 1 I=1,NENPT
    1 WRITE(6,101)I,(ENPT(J,I),J=1,4),(ENPT(J,I),J=6,8),(ENPT(J,I),
     . J=10,15)
   11 IF (NENAR.EQ.0)GO TO 12
      WRITE(6,110)SORNAM(1)
      DO 2 I=1,NENAR
    2 WRITE(6,111)I,(ENAR(J,I),J=1,4),(ENAR(J,I),J=6,10)
   12 IF(NENLN.EQ.0)GO TO 13
      WRITE(6,1200)  SORNAM(1)
      DO 5 I=1,NENLN
      WRITE(6,1211)  I,(ENLN(J,I),J=1,4),(ENLN(J,I),J=9,13)
    5 WRITE(6,1222)  (ENLN(J,I),J=6,8)
   13 IF (NABPT.EQ.0)GO TO 14
      WRITE(6,100)SORNAM(2)
      DO 3 I=1,NABPT
    3 WRITE(6,101)I,(ABPT(J,I),J=1,4),(ABPT(J,I),J=6,8),(ABPT(J,I),
     . J=10,15)
   14 IF(NABAR.EQ.0)GO TO 15
      WRITE(6,110)SORNAM(2)
      DO 4 I=1,NABAR
    4 WRITE(6,111)I,(ABAR(J,I),J=1,4),(ABAR(J,I),J=6,10)
   15 IF(NABLN.EQ.0)GO TO 16
      WRITE(6,1200)  SORNAM(2)
      DO 6 I=1,NABLN
      WRITE(6,1211)  I,(ABLN(J,I),J=1,4),(ABLN(J,I),J=9,13)
    6 WRITE(6,1222)  (ABLN(J,I),J=6,8)
   16 CONTINUE
  100 FORMAT(1H0,A8,14H POINT SOURCES/1X,119(1H-)/
     . 8X,1HI,11X,8HGEOMETRY,11X,22HI  STACK PARAMETERS  I,4X,1HI/
     . 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,3X,12HWIDTH I TEMP,4X,3HVEL,
     . 3X,11HDIAM I PR I,13X,28HEMISSIONS(MICROGRAMS/SECOND)/
     . 1X,14HNUMBER I  (KM),5X,4H(KM),4X,25H(M)    (M)  I(DEG K)  (M/S),
     . 3X,10H(M) IFLAGI,4X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/
     . 1X,119(1H-))
  101 FORMAT(I6,1X,2F9.2,2F7.1,F7.0,2F7.1,F4.0,5(1PE11.3))
  110 FORMAT(1H0,A8,13H AREA SOURCES/1X,94(1H-)/
     . 8X,1HI,11X,8HGEOMETRY,11X,1HI/
     . 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,4X,6HSIDE I,
     . 14X,28HEMISSIONS(MICROGRAMS/SECOND)/
     . 1X,14HNUMBER I  (KM),5X,4H(KM),4X,3H(M),4X,5H(M) I,
     . 5X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/1X,94(1H-))
  111 FORMAT(I6,1X,2F9.2,2F7.1,5(1PE11.3))
```

```
  120 FORMAT(1H0,A8,13H LINE SOURCES/1X,123(1H-)/
     . 8X,1HI,11X,8HGEOMETRY,11X,1HI,10X,1HI,54X,15HI AIRCRAFT ONLY/
     . 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,3X,18HWIDTH I VELOCITY I,
     . 13X,28HEMISSIONS(MICROGRAMS/SECOND),13X,15HI LENGTH   TIME/
     . 1X,14HNUMBER I  (KM),5X,4H(KM),4X,23H(M)     (M)   I (KM/HR)   I,
     . 5X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2,3X,15HI   (KM)     (HR)/
     . 1X,123(1H-))
  121 FORMAT(I6,1X,2F9.2,2F7.1,6(1PE11.3),0PF7.2,1PE11.3)
  122 FORMAT(7X,2F9.2,F7.1,7X,1PE11.3)
 1200 FORMAT(1H0,A8,13H LINE SOURCES/1X,96(1H-)/
     . 8X,1HI,11X,8HGEOMETRY,12X,1HI/
     . 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,4X,7HWIDTH I,
     . 23X,28HEMISSIONS(MICROGRAMS/SECOND)/
     . 1X,14HNUMBER I  (KM),5X,4H(KM),4X,3H(M),4X,6H(M)   I,
     . 6X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/
     . 1X,96(1H-))
 1211 FORMAT(I6,1X,2F9.2,2F7.1,2X,5(1PE11.3))
 1222 FORMAT(7X,2F9.2,F7.1)
  200 FORMAT(25H0WIND INDEPENDENT SOURCES/1H0)
      RETURN
C
      ENTRY DEPINP
C
C     AT THIS ENTRY ALL WIND DEPENDENT SOURCES ARE PRINTED
C
      WRITE(6,300)WS,WD
  300 FORMAT(1H1,'WIND DEPENDENT SOURCES FOR',F8.4,' MPS WIND SPEED AND'
     . ,F8.4,' RADIANS WIND DIRECTION')
      IF (IFLAG.EQ.0) GO TO 18
      IF (NACPT.EQ.0) GO TO 17
      WRITE(6,100) SORNAM(3)
      DO 7 I=1,NACPT
    7 WRITE(6,101)I,(ACPT(J,I),J=1,4),(ACPT(J,I),J=6,8),(ACPT(J,I),
     . J=11,15)
   17 IF(NACAR.EQ.0) GO TO 18
      WRITE(6,110) SORNAM(3)
      DO 8 I=1,NACAR
    8 WRITE(6,111)I,(ACAR(J,I),J=1,4),(ACAR(J,I),J=6,10)
   18 IF(NACLN.EQ.0) GO TO 19
      WRITE(6,120) SORNAM(3)
      DO 9 I=1,NACLN
      IF (ACLN(9,I).NE.1.0) GO TO 1987
      WRITE(6,1219) I,(ACLN(J,I),J=1,4),(ACLN(J,I),J=13,17),ACLN(11,I)
      WRITE(6,1229) (ACLN(J,I),J=6,8)
 1219 FORMAT(I6,1X,2F9.2,2F7.1,4X,3HN/A,4X,5(1PE11.3),0PF7.1,4X,3HN/A)
 1229 FORMAT(7X,2F9.2,F7.1,11X,3HN/A)
      GO TO 9
 1987 CONTINUE
      WRITE(6,121)I,(ACLN(J,I),J=1,4),ACLN(9,I),(ACLN(J,I),J=13,17),
     . ACLN(11,I),ACLN(12,I)
      WRITE(6,122)  (ACLN(J,I),J=6,8),ACLN(10,I)
    9 CONTINUE
   19 CONTINUE
      RETURN
      END
```

PROGRAM MAIN

Purpose:

To read the general problem input, set up the receptor grid, call a routine to read the master emission file and then call the short-term model.

Input:

1. Problem title
2. Definition of pollutants to be output
3. Description of special cases
4. Description of receptor grid
5. Description of statistical receptors

Output:

All input is printed

Procedure:

1. Read card input
2. Calculate receptor locations
3. Check statistical receptors against the receptor locations
4. Call routine to read master emission file
5. Call the short-term model

Subroutines Called:

READ, MAINS

AD-A046 348 ARGONNE NATIONAL LAB ILL F/G 13/2

AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)

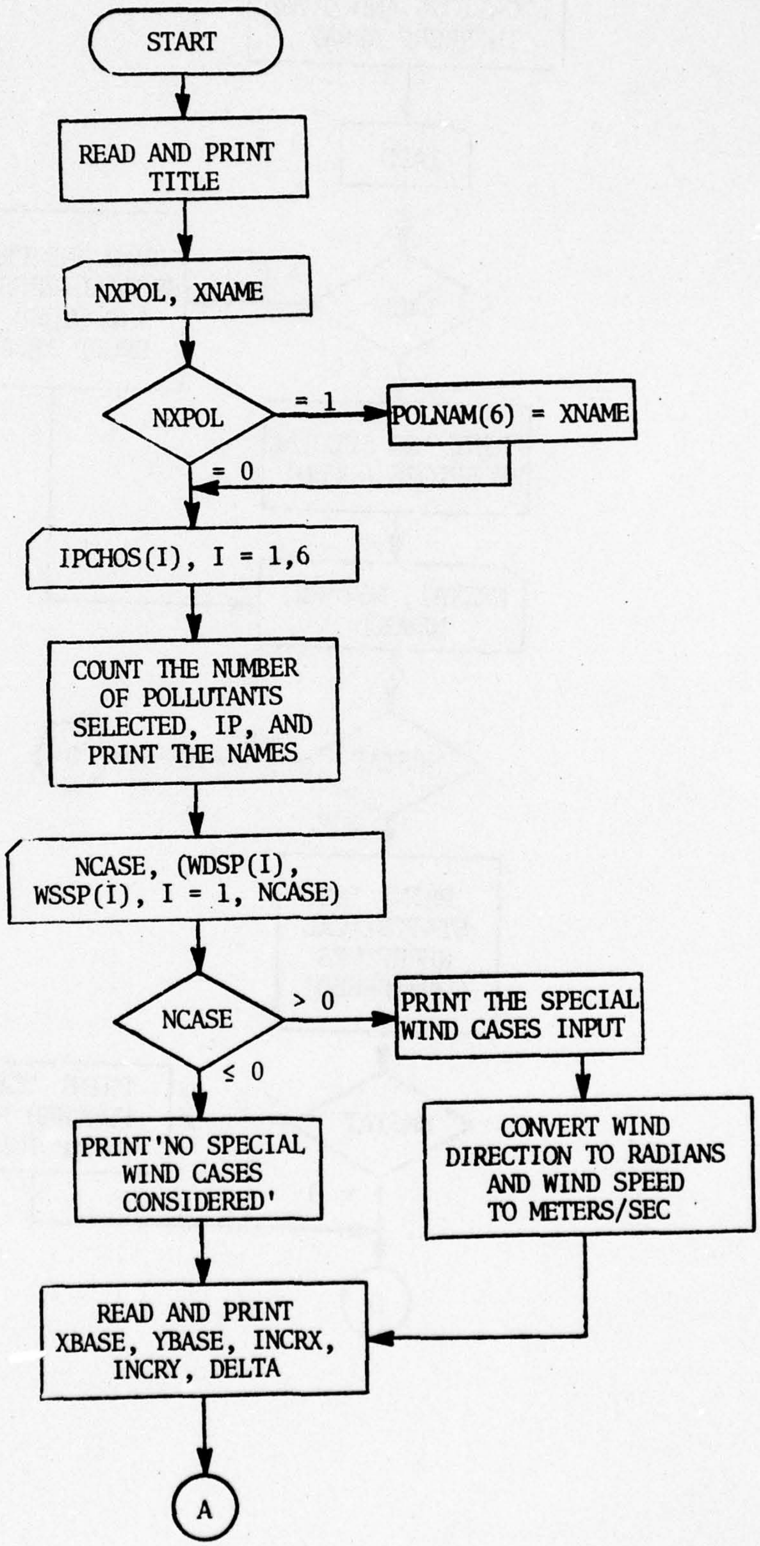
APR 77 D J BINGAMAN

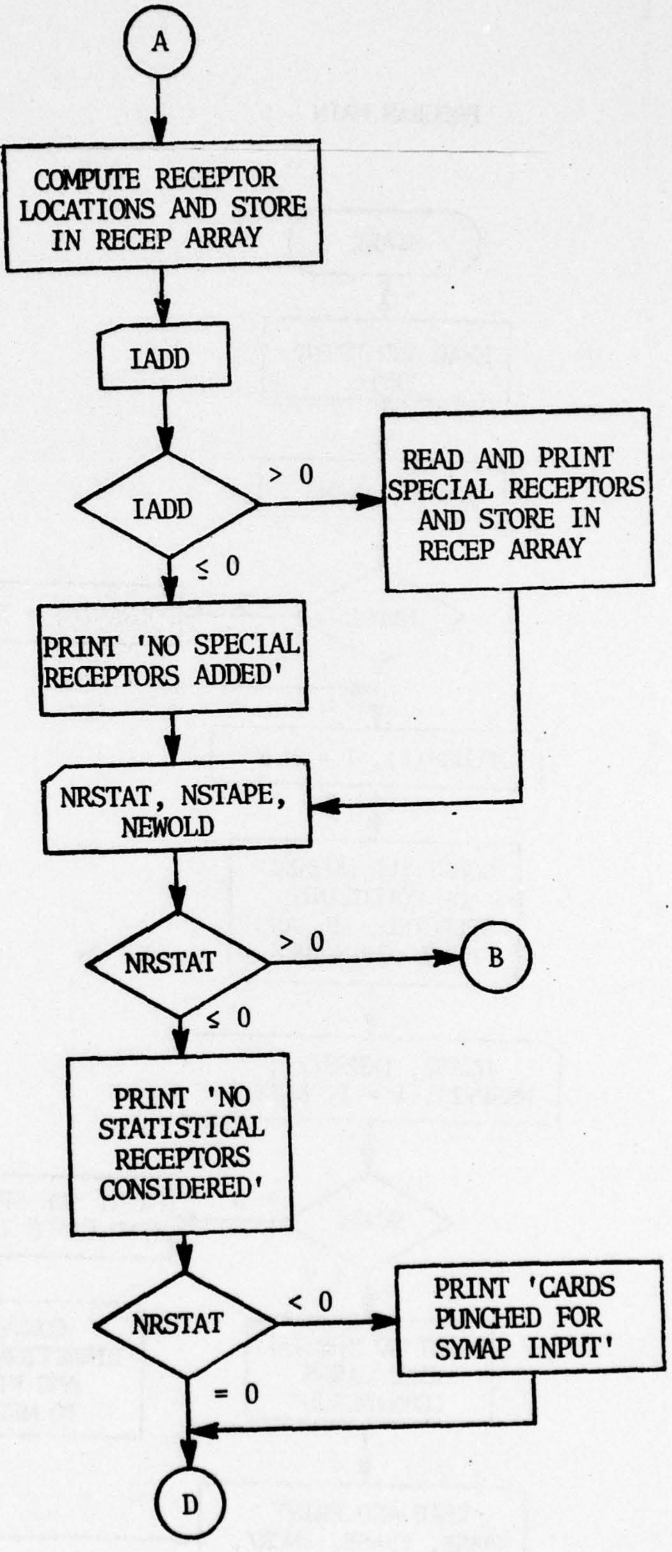
UNCLASSIFIED CEEDO-TR-76-34 NL

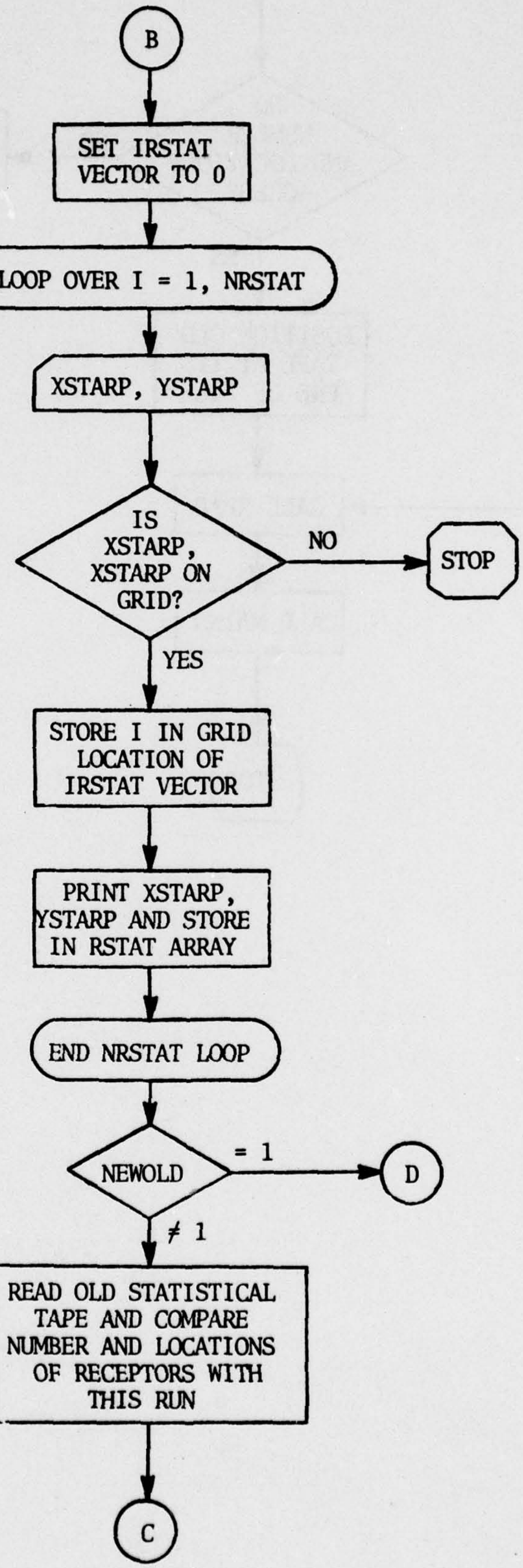
2 OF 3

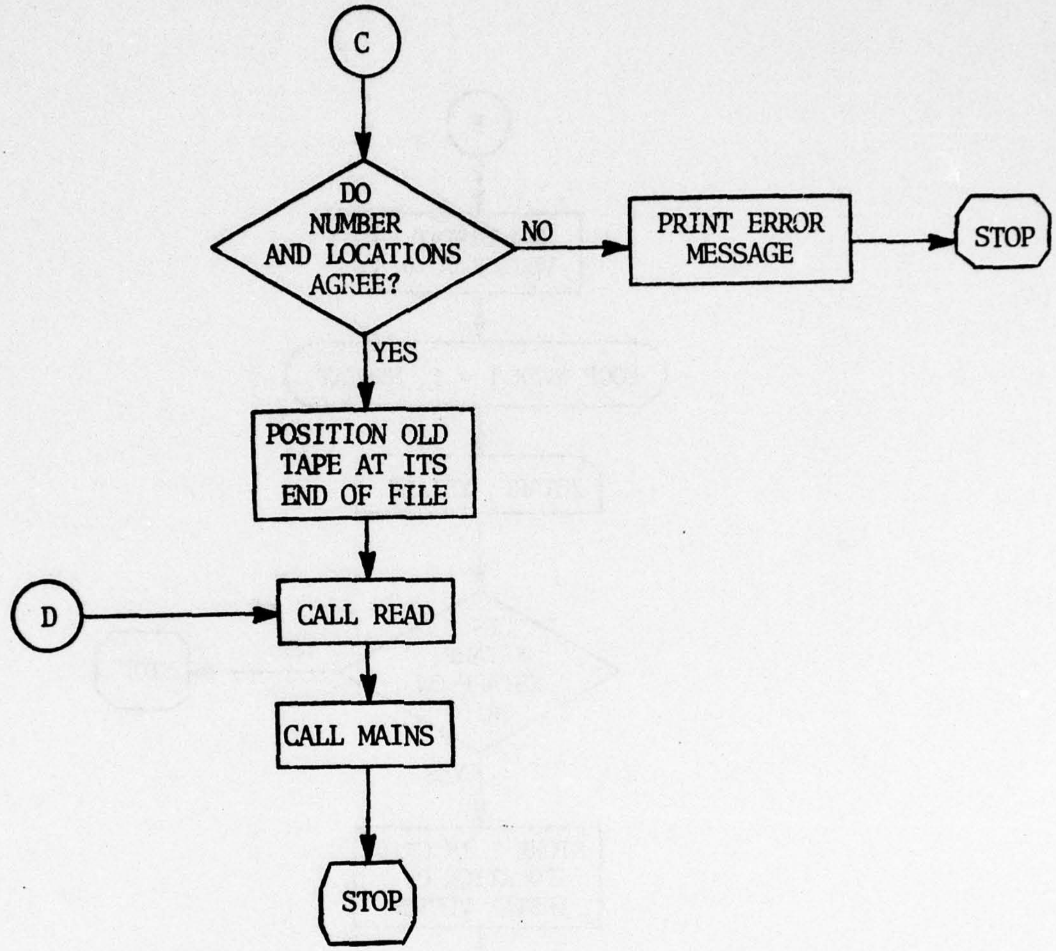
AD A046348

PROGRAM MAIN




A
COMPUTE RECEPTOR LOCATIONS AND STORE IN RECEP ARRAY
IADD
IADD
> 0
READ AND PRINT SPECIAL RECEPTORS AND STORE IN RECEP ARRAY
≤ 0
PRINT 'NO SPECIAL RECEPTORS ADDED'
NRSTAT, NSTAPE, NEWOLD
NRSTAT
> 0
B
≤ 0
PRINT 'NO STATISTICAL RECEPTORS CONSIDERED'
NRSTAT
< 0
PRINT 'CARDS PUNCHED FOR SYMAP INPUT'
= 0
D


B
SET IRSTAT VECTOR TO 0
LOOP OVER I = 1, NRSTAT
XSTARP, YSTARP
IS XSTARP, XSTARP ON GRID?
NO
STOP
YES
STORE I IN GRID LOCATION OF IRSTAT VECTOR
PRINT XSTARP, YSTARP AND STORE IN RSTAT ARRAY
END NRSTAT LOOP
NEWOLD
= 1
D
≠ 1
READ OLD STATISTICAL TAPE AND COMPARE NUMBER AND LOCATIONS OF RECEPTORS WITH THIS RUN
C


C
DO NUMBER AND LOCATIONS AGREE?
NO
PRINT ERROR MESSAGE
STOP
YES
POSITION OLD TAPE AT ITS END OF FILE
D
CALL READ
CALL MAINS
STOP

```
C
C     THIS PROGRAM IS THE MAIN DRIVER ROUTINE WHICH READS IN RECEPTOR AND
C     OTHER GENERAL DATA, CALLS SUBROUTINE READ TO READ THE MASTER
C     SOURCE EMISSION TAPE, AND THE DIRECTS CONTROL TO MAINS FOR THE
C     SHORT TERM MODEL
C
      REAL*8 POLNAM,XNAME
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,SORC(18),IPOL
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP
     . ,TEMK,UA
      COMMON /MONMET/ TMBAR,WSMBAR,AMDMBR,DTMBAR
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /SPEC/ NCASE,WSSP(3),WDSP(3)
      COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      COMMON /TITL/ POLNAM(6),TITLE1(20),IPCHOS(6),NXPOL,IP
      COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)
      DIMENSION AA (20),OSTATR(2,20)
C
C     READ AND PRINT RECEPTOR AND OTHER GENERAL INPUT
C
    1 READ(5,100) TITLE1
  100 FORMAT(20A4)
      PRINT 200, TITLE1
  200 FORMAT(1H1,20A4)
      READ(5,110)       NXPOL, XNAME
  110 FORMAT( I6,5A8)
      IF(NXPOL.EQ.0)GO TO 31
      POLNAM(6 )=XNAME
   31 CONTINUE
      READ(5,130) (IPCHOS(I),I=1,6)
  130 FORMAT(10I6)
      DO 40 I=1,6
      IF(IPCHOS(I).LE.0) GO TO 41
   40 CONTINUE
   41 IP=I-1
      PRINT 203, (POLNAM(IPCHOS(I)),I=1,IP)
  203 FORMAT(21H0POLLUTANTS SELECTED /6A8)
      READ(5,140)NCASE,(WDSP(I),WSSP(I),I=1,NCASE)
  140 FORMAT(I6,6F6.0)
      IF (NCASE) 48,48,49
   48 PRINT 201
  201 FORMAT(33H0NO SPECIAL WIND CASES CONSIDERED)
      GO TO 51
   49 PRINT 202,       (I,WDSP(I),WSSP(I),I=1,NCASE)
  202 FORMAT(20H0SPECIAL WIND CASES /53H   CASE   WIND DIRECTION(DEGREES
     .)    WIND SPEED(KNOTS)/(I6,F18.2,F23.2))
      DO 50 I=1,NCASE
      WDSP(I)=WDSP(I)* 0.0174533
   50 WSSP(I)=WSSP(I)*0.5148
   51 CONTINUE
      READ(5,120)XBASE,YBASE,INCRX,INCRY,DELTA
  120 FORMAT(2F8.0,2I8,F8.0)
      PRINT 204, XBASE,YBASE,INCRX,INCRY,DELTA
  204 FORMAT(43H0LOWER LEFT CORNER OF RECEPTOR GRID IS AT (,F8.3,1H,,
     .F8.3,1H)/12H   THERE ARE,I4,12H COLUMNS AND,I4,23H ROWS WITH A SPA
     .CING OF,F6.2,11H KILOMETERS)
      NRECEP=0
```

```
      DO 10 I=1,INCPX
      DO 10 J=1,INCRY
      NRECEP=NRECEP+1
      RECEP(1,NRECEP)=XBASE+(I-1)*DELTA
   10 RECEP(2,NRECEP)=YBASE+(J-1)*DELTA
      READ(5,110)IADD
      IF (IADD) 14,14,15
   14 PRINT 205
  205 FORMAT(27HONO SPECIAL RECEPTORS ADDED)
      GO TO 21
   15 PRINT 206
  206 FORMAT(25HOSPECIAL RECEPTORS ADDED /36H    NO.    X-COORDINATE    Y-C
     .OORDINATE)
      DO 20 I=1,IADD
      READ(5,120)XRECEP,YRECEP
      NRECEP=NRECEP+1
      PRINT 207, NRECEP,XRECEP,YRECEP
  207 FORMAT(I5,F14.3,F15.3)
      RECEP(1,NRECEP)=XRECEP
      RECEP(2,NRECEP)=YRECEP
   20 CONTINUE
   21 CONTINUE
C
C   READ AND PRINT STATISTICAL RECEPTOR INPUT
C
      READ 130,NRSTAT,NSTAPE,NEWOLD
      IF (NRSTAT.GT.0) GO TO 305
      PRINT 302
  302 FORMAT(36HONO STATISTICAL RECEPTORS CONSIDERED)
      IF (NRSTAT.LT.0) PRINT 303
  303 FORMAT(30HOCARDS PUNCHED FOR SYMAP INPUT)
      GO TO 400
  305 PRINT 301,NRSTAT
  301 FORMAT(1H0,I8,22H STATISTICAL RECEPTORS)
      DO 310 I=1,NRECEP
      IRSTAT(I)=0
  310 CONTINUE
      DO 340 I=1,NRSTAT
      READ 120,XSTARP,YSTARP
      DO 320 IC=1,NRECEP
      IF (XSTARP.EQ.RECEP(1,IC).AND.YSTARP.EQ.RECEP(2,IC)) GO TO 330
  320 CONTINUE
      PRINT 321,XSTARP,YSTARP
  321 FORMAT(25HOSTATISTICAL RECEPTOR X =,F7.3,5H, Y =,F7.3,
     . 12H NOT ON GRID)
      STOP
  330 IRSTAT(IC)=I
      PRINT 322,I,XSTARP,YSTARP
  322 FORMAT(I12,7H AT X =,F10.3,4H Y =,F10.3)
      RSTAT(1,I)=XSTARP
      RSTAT(2,I)=YSTARP
  340 CONTINUE
      IF (NEWOLD.EQ.1) GO TO 400
      READ (NSTAPE) AA,IH,JRSTAT,((OSTATR(I,J),I=1,2),J=1,JRSTAT),
     . JWD,WDJ,JWS,WSJ,JJJ,HLIDJ
      IF (JRSTAT.EQ.NRSTAT) GO TO 350
      PRINT 341
  341 FORMAT(46HONUMBER OF STATISTICAL RECEPTORS ON OLD TAPE (,I2,
     . 42H), DO NOT AGREE WITH NUMBER FOR THIS RUN (,I2,1H))
      STOP
  350 DO 360 J=1,NRSTAT
      DO 360 I=1,2
```

```
      IF (OSTATR(I,J).EQ.RSTAT(I,J)) GO TO 360
      PRINT 351,((OSTATR(I,J),I=1,2),J=1,NRSTAT)
  351 FORMAT(64HOSTATISTICAL RECEPTOR LOCATIONS DO NOT MATCH,THOSE ON TA
     .PE ARE ,20(/2F15.3))
      STOP
  360 CONTINUE
C
C   POSITION OLD TAPE AT END OF LAST RECORD
C
  370 READ (NSTAPE,END=380)
      GO TO 370
  380 BACKSPACE NSTAPE
      BACKSPACE NSTAPE
  400 CONTINUE
      CALL  READ
      CALL MAINS
      STOP
      RETURN
      END
```

SUBROUTINE MAINS

Purpose:

To direct the short term calculation by reading the data, checking for special cases and calling the source and diffusion calculation routines.

Input:

1. Card input to describe the time periods to be calculated
2. Card input to describe the meteorological conditions
3. Special case data from the MAIN routine

Output:

1. Common block data to be used by the calculation and output subroutines
2. Statistical receptor data on tape and/or cards for SYMAP

Procedure:

1. Set constants
2. Read general data
3. For each period:
   a. Read time and meteorological data
   b. Check for near zero wind speed
   c. Check limits of mixing depth
   d. For each hour:
      1. Find the critical distance
      2. Set wind direction and speed classes
      3. Call the non-aircraft source routines
      4. Check for special cases of wind speed and direction
      5. Call the aircraft source routines
      6. Call the diffusion model and output routines
      7. Check for statistical output, including cards for SYMAP.

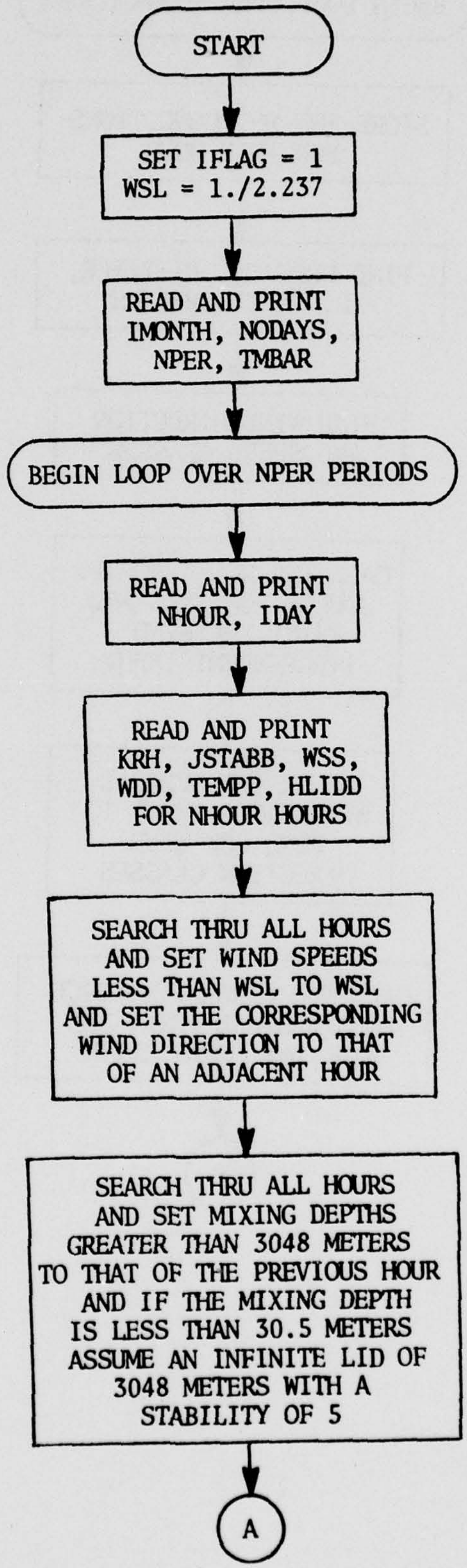
Subroutines Called:

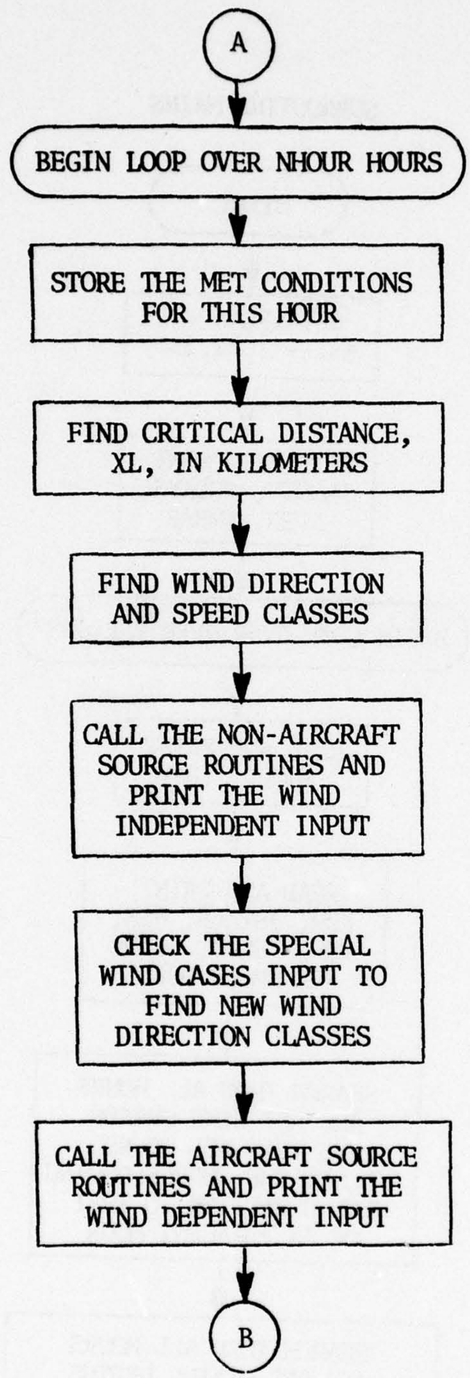
SOURCE, INDINP, ACSRCE, DEPINP, POLSOR, OUTPUT

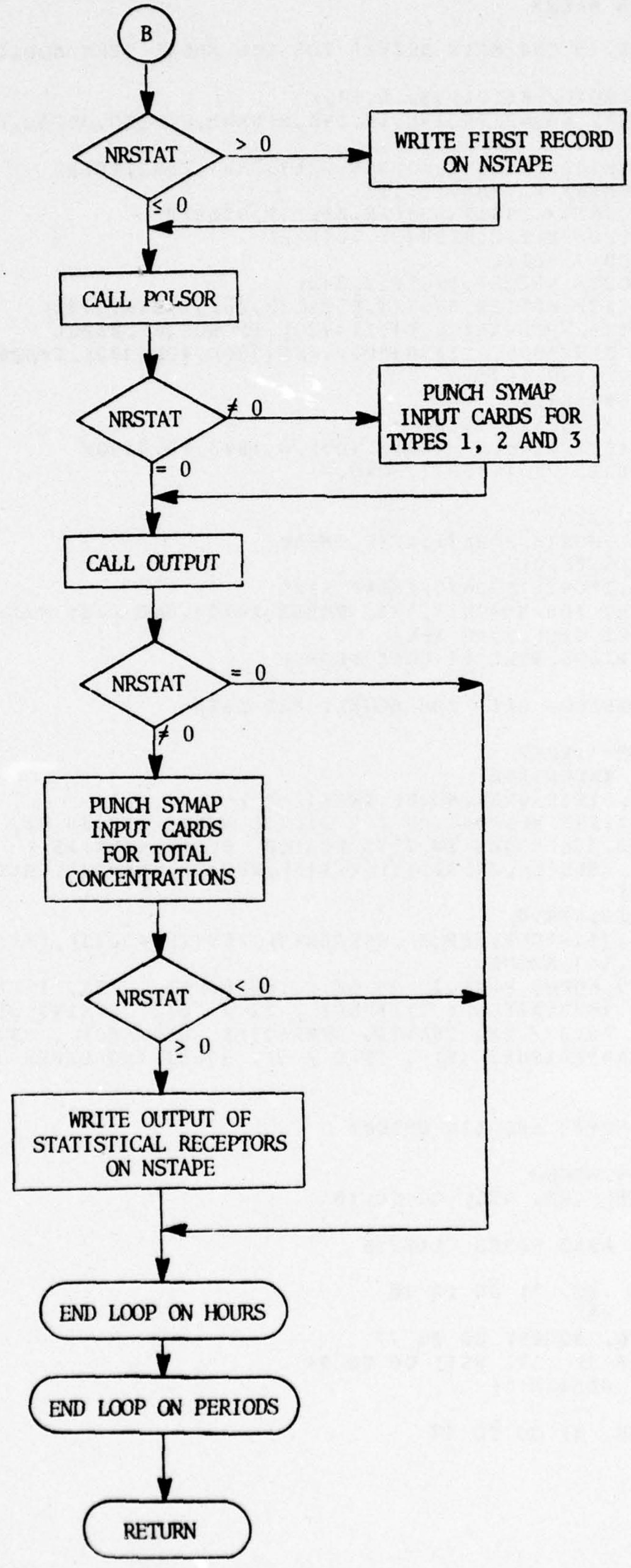
Function Called:

SIGCZ

SUBROUTINE MAINS




A
BEGIN LOOP OVER NHOUR HOURS
STORE THE MET CONDITIONS FOR THIS HOUR
FIND CRITICAL DISTANCE, XL, IN KILOMETERS
FIND WIND DIRECTION AND SPEED CLASSES
CALL THE NON-AIRCRAFT SOURCE ROUTINES AND PRINT THE WIND INDEPENDENT INPUT
CHECK THE SPECIAL WIND CASES INPUT TO FIND NEW WIND DIRECTION CLASSES
CALL THE AIRCRAFT SOURCE ROUTINES AND PRINT THE WIND DEPENDENT INPUT
B


B
NRSTAT
> 0
WRITE FIRST RECORD ON NSTAPE
≤ 0
CALL POLSOR
NRSTAT
≠ 0
PUNCH SYMAP INPUT CARDS FOR TYPES 1, 2 AND 3
= 0
CALL OUTPUT
NRSTAT
= 0
≠ 0
PUNCH SYMAP INPUT CARDS FOR TOTAL CONCENTRATIONS
NRSTAT
< 0
≥ 0
WRITE OUTPUT OF STATISTICAL RECEPTORS ON NSTAPE
END LOOP ON HOURS
END LOOP ON PERIODS
RETURN

```
      SUBROUTINE MAINS
C
C     THIS ROUTINE IS THE MAIN DRIVER FOR THE SHORT TERM MODEL
C
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG
      COMMON /XTRAN/ XL,WSMD,TY,TZ
      COMMON /MONMET/ TMBAR,WSMBAR,AMDMBR,DTMBAR
      COMMON /SPEC/ NCASE,WSSP(3),WDSP(3)
      COMMON /WDUN/ WSAVE
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)
      COMMON /TITL/ POLNAM(6),TITLE1(20),IPCHOS(6),NXPOL
      DIMENSION KRH(100),JSTABB(100),WSS(100),WDD(100),TEMPP(100),
     . HLIDD(100),WSCLAS(5)
      DIMENSION WEEK(2)
      DATA WEEK / 3HDAY, 3HEND /
      DATA WSCLAS/1.8018,3.3462,5.4055,8.4943,10.8110/
      DATA PI,SHLID /3.1415927,3048./
      IFLAG = 1
      WSL = 1./2.237
      READ 100, IMONTH,NODAYS,NPER,TMBAR
  100 FORMAT (3I6,F6.0)
      PRINT 203,IMONTH,NODAYS,TMBAR,NPER
  203 FORMAT(11H0 FOR MONTHI3,11H, THERE AREI4,36H DAYS WITH AN AVERAGE
     .TEMPERATURE OFF6.2,4H (F)/
     .I5,27H PERIODS WILL BE CONSIDERED)
C
C     FOR EVERY PERIOD, READ THE HOURLY MET DATA
C
      DO 50 IPER=1,NPER
      READ 100, NHOUR,IDAY
      PRINT 201, IPER,NPER,NHOUR,WEEK(IDAY)
  201 FORMAT(1H1,29HINFORMATION FOR PERIOD NUMBER,I3,3H OF, I3,8H PERIOD
     .S  / 5X,I3,33H HOURS IN THIS PERIOD, FOR A WEEK,A3 )
      READ 101, (KRH(I),JSTABB(I),WSS(I),WDD(I),TEMPP(I),HLIDD(I),
     -I=1,NHOUR)
  101 FORMAT (2I6,4F6.0)
      PRINT 202,(I,NHOUR,KRH(I),JSTABB(I),WSS(I),WDD(I),TEMPP(I),
     . HLIDD(I),I=1,NHOUR)
  202 FORMAT(1H0,8HFOR HOUR,I5,3H OF,I5,6H HOURS   /5X, 10HHOUR INDEX ,
     . I3 / 5X, 18HSTABILITY CATEGORY , I2 /  5X, 26HWIND SPEED (METERS/
     .SECOND) , F8.2 / 5X, 24HWIND DIRECTION (DEGREES) , F8.2 /
     . 5X, 15HTEMPERATURE (F) , F8.2 / 5X, 21HMIXING DEPTH (METERS) ,
     . F 8.2 )
C
C     CHECK WIND SPEED AND LID HEIGHT
C
      DO 20 IH=1,NHOUR
      IF (WSS(IH) .GE. WSL) GO TO 18
C
C     REMOVE ZERO WIND SPEED CLASSES
C
      IF (NHOUR .EQ. 1) GO TO 16
      WSS(IH) = WSL
      IF (IH .EQ. NHOUR) GO TO 17
      IF (WSS(IH+1) .LT. WSL) GO TO 14
      WDD(IH) = WDD(IH+1)
      GO TO 18
   14 IF (IH .NE. 1) GO TO 17
```

```
      DO 15 JH=2,NHOUR
      IF (WSS(JH) .LT. WSL) GO TO 15
      WDD(IH)= WDD(JH)
      GO TO 18
   15 CONTINUE
   16 PRINT 200
      STOP
  200 FORMAT(1H1,10X,'WINDSPEED IS TOO SMALL')
   17 WDD(IH) = WDD(IH-1)
C
C  SET INFINITE LID HEIGHT AT 3048 METERS
C
   18 IF(HLIDD(IH).GT. 3048.) HLIDD(IH) = SHLID
      IF (HLIDD(IH) .GE. 30.5) GO TO 19
C
C  GROUND LEVEL INVERSION, ASSUME INFINITE LID WITH A STABILTY OF 5
C
      HLIDD(IH) = 3048.
      JSTABB(IH) = 5
   19 SHLID = HLIDD(IH)
   20 CONTINUE
C
C  BEGIN HOURLY CALCULATIONS
C
      DO 30 IHOUR = 1,NHOUR
      IHR1 = KRH(IHOUR)
      WS = WSS(IHOUR)
      IHR2 = IHR1
      WD = WDD(IHOUR)*0.0174533
      TEMF = TEMPP(IHOUR)
      TEMK = (TEMF-32.)/1.8 + 273.
      JSTAB = JSTABB(IHOUR)
      HLID = HLIDD(IHOUR)
      WSMPH = WS * 2.237
      SINEWD = SIN(WD)
      COSEWD = COS(WD)
      HL = 0.47*HLID
      WSAVE=WS
      IF(HL.LT.1.0)HL=1.0
C
C   FIND CRITICAL DISTANCE, XL, IN KILOMETERS
C
      XL=SIGCZ(JSTAB,HL)/1000.
C
C   FIND WIND DIRECTION AND SPEED CLASSES
C
      DO 21 K=1,16
      IF (WD .GT. (22.5*K-11.25)*PI/180.) GO TO 21
      IWD = K
      GO TO 22
   21 CONTINUE
      IWD=1
   22 DO 23 K=1,5
      IF (WS.GT.WSCLAS(K)) GO TO 23
      IWS = K
      GO TO 25
   23 CONTINUE
      IWS = 6
   25 CONTINUE
      PRINT 925,IHOUR
  925 FORMAT(9H1FOR HOURI3)
C
```

```
C   CALL THE NON-AIRCRAFT SOURCE ROUTINES AND PRINT THE
C   WIND INDEPENDENT INPUT
C
      CALL SOURCE
      CALL INDINP
      IWDT = IWD
      IF (NCASE .EQ. 0) GO TO 28
      DO 26 I=1,NCASE
      COMP = WS * COS(WD - WDSP(I))
      IF (COMP .LT. WSSP(I)) GO TO 26
      IWD = 17 + I
      GO TO 28
   26 CONTINUE
   28 CONTINUE
C
C   CALL THE AIRCRAFT SOURCE ROUTINES AND PRINT THE
C   WIND DEPENDENT INPUT
C
       CALL ACSRCE
      CALL DEPINP
      IWD = IWDT
C
C   IF STATISICAL OPTION IS CHOSEN, WRITE FIRST RECORD ON NSTAPE
C
      IF (NRSTAT.LE.0) GO TO 300
      NHR=1
      WRITE (NSTAPE) TITLE1,IHR1,NRSTAT,((RSTAT(I,J),I=1,2),J=1,NRSTAT),
     . IWD,WD,IWS,WS,JSTAB,HLID,NHR
  300 CONTINUE
C
C   CALL THE DIFFUSION MODEL
C
      CALL PCLSOR
      IF (NRSTAT.EQ.0) GO TO 320
      DO 310 K=1,3
      PUNCH 301,NRECEP,K
  301 FORMAT(2I6)
      DO 310 N=1,NRECEP
      PUNCH 302,(RECEP(I,N),I=1,2),(RECDAT(K,J,N),J=1,6)
  302 FORMAT(1P8E10.3)
  310 CONTINUE
  320 CONTINUE
C
C    PRINT RESULTS
C
      CALL OUTPUT
C
C    IF THE STATISTICAL OPTION IS CHOSEN, RECORD THE OUTPUT
C
      IF (NRSTAT.EQ.0) GO TO 360
      K=4
      PUNCH 301,NRECEP,K
      DO 330 N=1,NRECEP
      PUNCH 302,(RECEP(I,N),I=1,2),(RECDAT(1,J,N),J=1,6)
  330 CONTINUE
      IF (NRSTAT.LT.0) GO TO 360
      K=0
      DO 350 N=1,NRECEP
      IF (IRSTAT(N).LE.0) GO TO 350
      K=K+1
      DO 340 J=1,6
      RECDAT(1,J,K)=RECDAT(1,J,N)
```

```
340 CONTINUE                                                  MAINS186
350 CONTINUE                                                  MAINS187
    WRITE (NSTAPE) ((RECDAT(1,J,N),J=1,6),N=1,NRSTAT)         MAINS188
360 CONTINUE                                                  MAINS189
30  CONTINUE                                                  MAINS190
50  CONTINUE                                                  MAINS191
    RETURN                                                    MAINS192
    END                                                       MAINS193
```

SUBROUTINE METHA

Purpose:

To calculate diurnal emissions allowing each source in a class to have a unique or default distribution pattern.

Input:

1. The ICLASS number of the sources and NPTC, the number of sources not using the default of a uniform distribution.
2. For each of the NPTC sources, the source ID number and fractions of the hour, day and month, FH, FD and FM, which that source is on. If one or two of the fractions are left blank, the default is used. If all are blank, the source is assumed to be off.

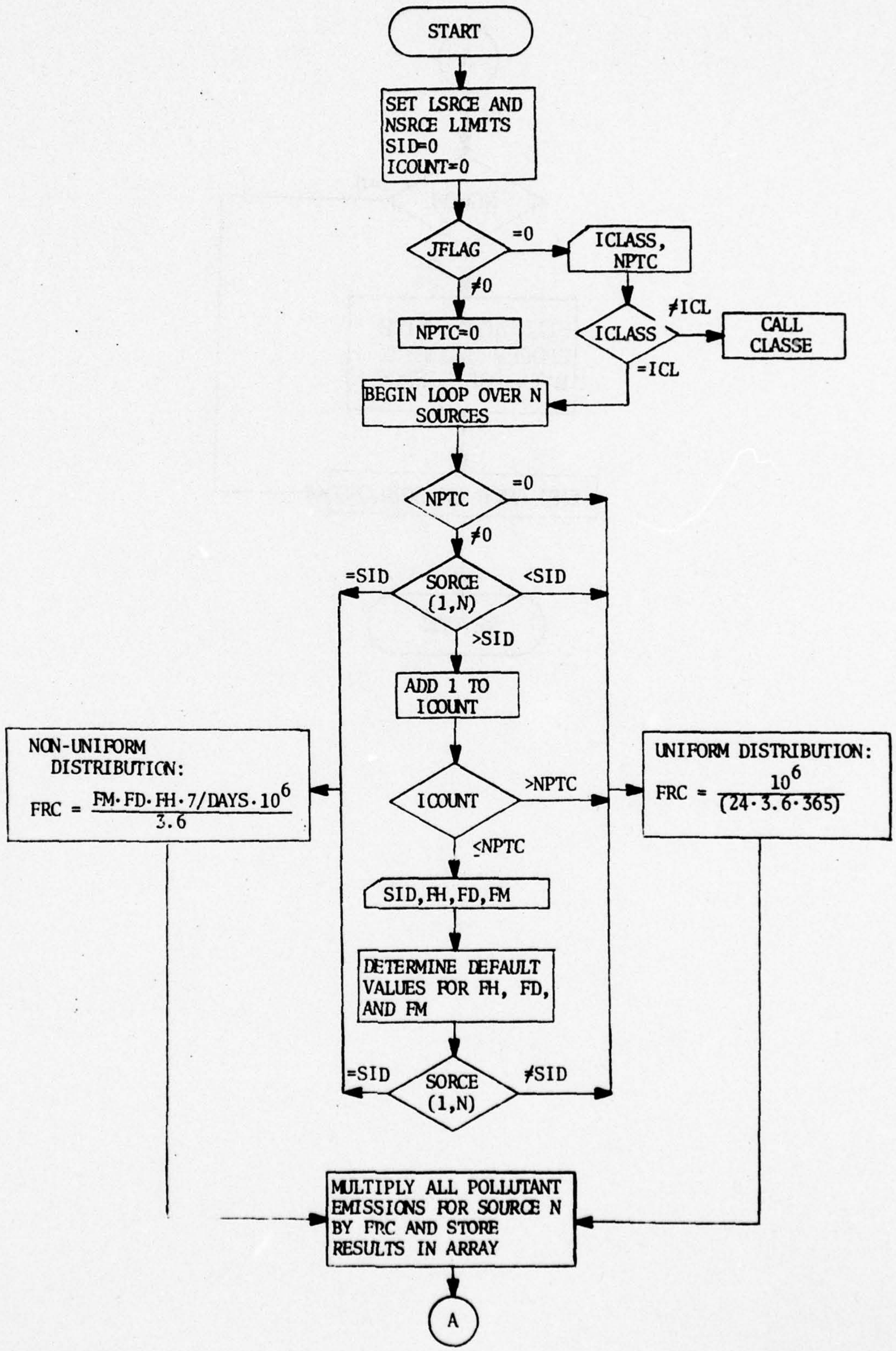
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

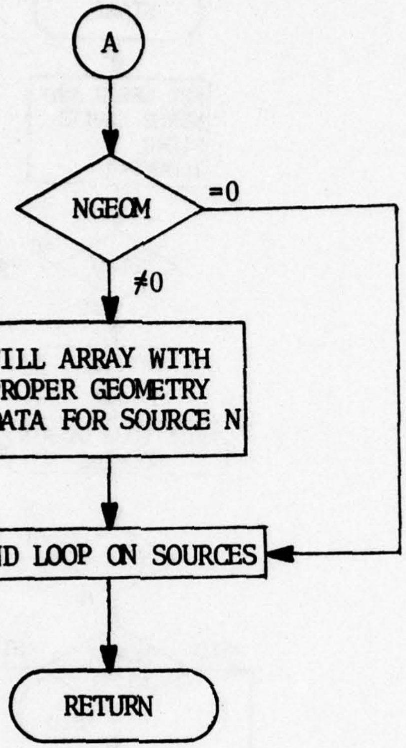
Subroutines Called:

CLASSE

SUBROUTINE METHA



SUBROUTINE METHA (Contd.)



```
      SUBROUTINE METHA(MAXN,ARRAY,I1,I2,ICL)
C
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS ALLOWING EACH
C     SOURCE IN A CLASS TO HAVE A DIFFERENT DISTRIBUTION PATTERN.
C     DEFAULTS ARE: FH = 1/24
C                   FD = 1/7
C                   FM = 1/12 OR 1
C
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,N[illegible]LN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                   ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DA[illegible]SRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NG[illegible]PT
      DIMENSION ARRAY([illegible]1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
      SID=0.
      ICOUNT=0
      IF (JFLAG.EQ.0) GO TO 5
      NPTC=0
      GO TO 6
    5 READ 1,ICLASS,NPTC
    1 FORMAT(2I4)
      IF(ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)
    6 DO 100 N=LSRCE,NSRCE
      IF (NPTC.EQ.0) GO TO 30
      IF (SID-SORCE(1,N))10,40,30
   10 ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.NPTC) GO TO 30
      READ 2,SID,FH,FD,FM
    2 FORMAT(F4.0,4X,3F8.7)
      IF (FH+FD+FM.EQ.0.0) GO TO 20
C
C     DETERMINE DEFAULT VALUES
C
      IF (FM.NE.0.0) GO TO 15
      FM=1./12.
      IF (DAYS.GE.365.) FM=1.
   15 IF (FD.EQ.0.0) FD=1./7.
      IF (FH.EQ.0.0) FH=1./24.
C
   20 CONTINUE
      IF (SID-SORCE(1,N))30,40,30
C
C     UNIFORM DISTRIBUTION
C
   30 FRC=1.0E+6/(24.*3.6*365.)
      GO TO 50
C
C     NON-UNIFORM DISTRIBUTION
C
   40 FRC=FM*FD*FH*(7./DAYS)*(1.0E+6/3.6)
   50 DO 60 I=1,NPLTS
      ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC
   60 CONTINUE
      IF (NGEOM.EQ.0) GO TO 100
      DO 70 I=1,NGEOM
      ARRAY(I,N)=SORCE(I+2,N)
   70 CONTINUE
      IF (IPT.EQ.1) ARRAY(10,N)=SORCE(2,N)
  100 CONTINUE
```

```
      RETURN                                                              METHA062
      END                                                                 METHA063
```

SUBROUTINE METHB

Purpose:

To calculate diurnal emissions using a degree-hour method.

Input:

The ICLASS number of the sources and UNIFRC, the fraction of emissions which are to be uniformly distributed.

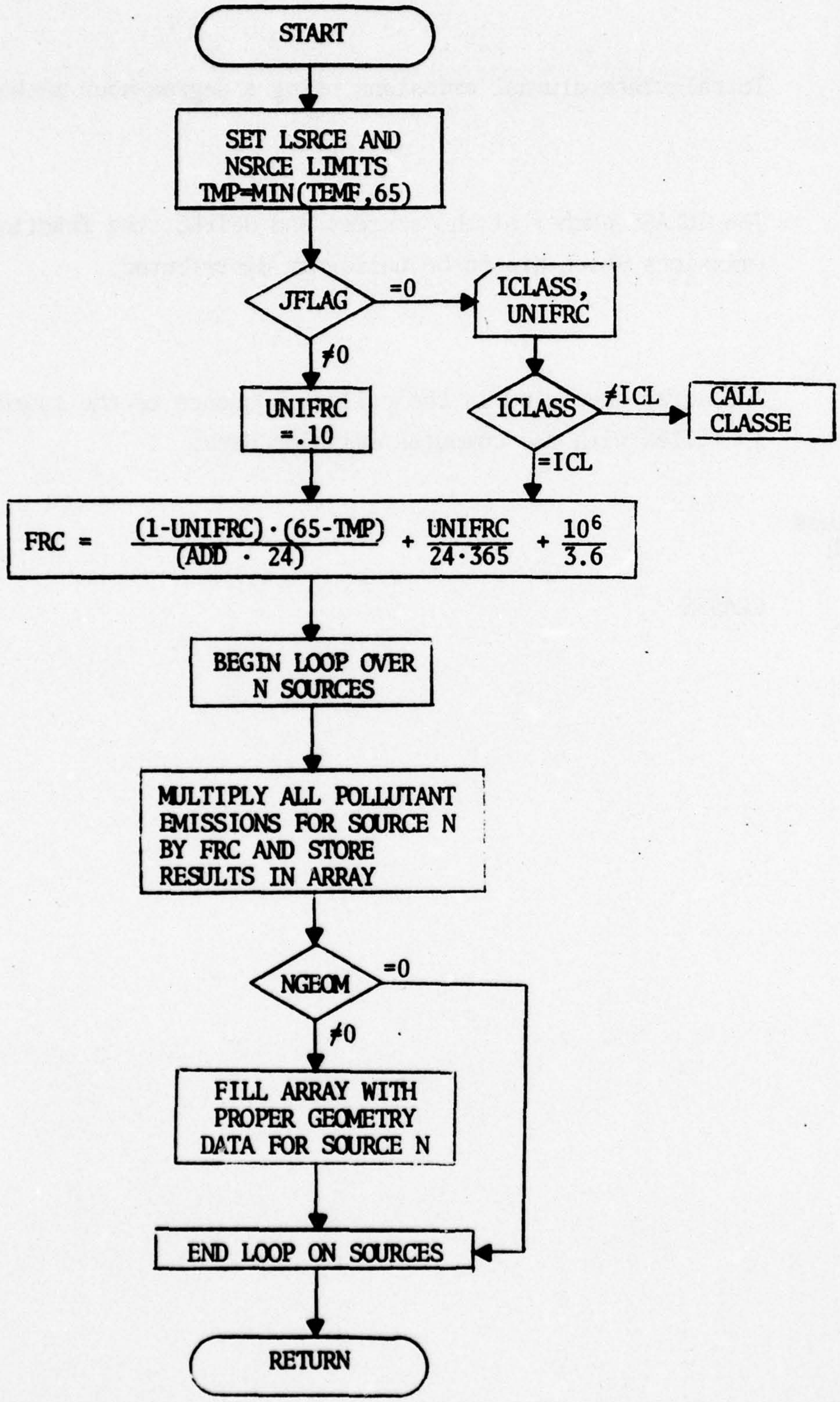
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines Called:

CLASSE

SUBROUTINE METHB



```
      SUBROUTINE METHB(MAXN,ARRAY,I1,I2,ICL)                            METHB000
C                                                                       METHB001
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING                   METHB002
C     A DEGREE-HOUR METHOD                                              METHB003
C                                                                       METHB004
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG          METHB005
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,          METHB006
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),         METHB007
     .                ABPT(16,150),ABAR(11,100),ABLN(14,100)            METHB008
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)          METHB009
     . ,LOC1,LOC2,NGEOM,IPT                                             METHB010
      COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,                       METHB011
     .  JSTAB,HLID,TEMP,TEMK                                            METHB012
      DIMENSION ARRAY(I1,I2)                                            METHB013
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR                         METHB014
      LSRCE=NSRCE+1                                                     METHB015
      NSRCE=NSRCE+MAXN                                                  METHB016
      TMP=TEMP                                                          METHB017
      IF (TEMP.GT.65.) TMP=65.                                          METHB018
      IF (JFLAG.EQ.0) GO TO 5                                           METHB019
      UNIFRC=.10                                                        METHB020
      GO TO 6                                                           METHB021
    5 READ 1,ICLASS,UNIFRC                                              METHB022
    1 FORMAT(I4,4X,F8.7)                                                METHB023
      IF(ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)                        METHB024
    6 FRC=(((1.0-UNIFRC)*((65.0-TMP)/(ADD*24.0)))+(UNIFRC/(24.0*365.0)))METHB025
     .    * (1.0E+6/3.6)                                                METHB026
C                                                                       METHB027
      DO 30 N=LSRCE,NSRCE                                               METHB028
      DO 10 I=1,NPLTS                                                   METHB029
      ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC                               METHB030
   10 CONTINUE                                                          METHB031
      IF (NGEOM.EQ.0) GO TO 30                                          METHB032
      DO 20 I=1,NGEOM                                                   METHB033
      ARRAY(I,N)=SORCE(I+2,N)                                           METHB034
   20 CONTINUE                                                          METHB035
   30 CONTINUE                                                          METHB036
      RETURN                                                            METHB037
      END                                                               METHB038
```

SUBROUTINE METHC

Purpose:

To calculate diurnal emissions using the same distribution pattern for all sources in the class.

Input:

The ICLASS number of the sources and the fractions of the hour, day and month, FH, FD and FM, which the sources are on. If one or two of the fractions are left blank, the default is used. If all are blank, the sources are assumed to be off.

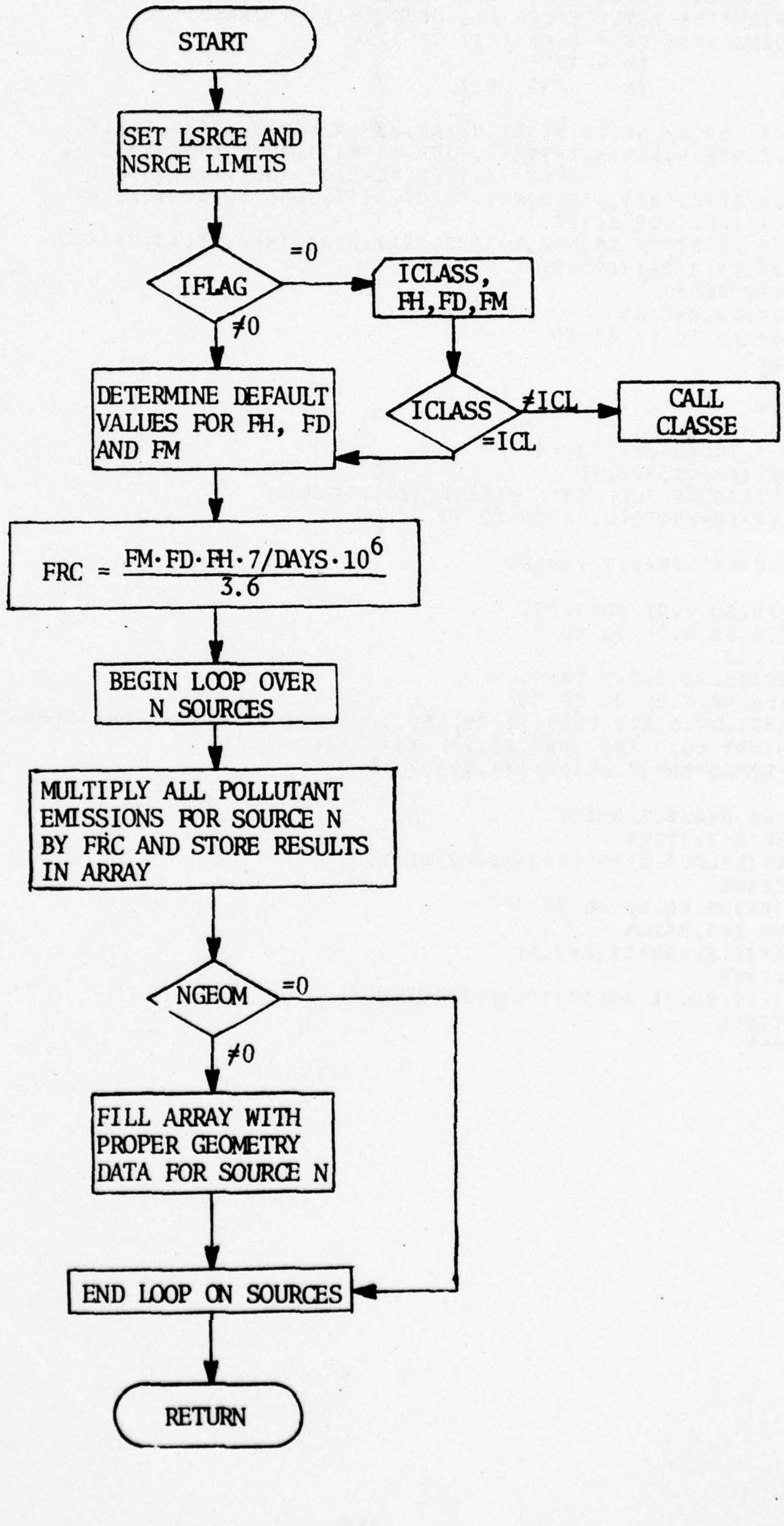
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines Called:

CLASSE

SUBROUTINE METHC



```
      SUBROUTINE METHC(MAXN,ARRAY,I1,I2,ICL)
C
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE SAME
C     DISTRIBUTION PATTERN FOR ALL SOURCES IN A CLASS.
C     DEFAULTS ARE: FH = 0 OR 1/12 OR 1/24
C                   FD = 1/7
C                   FM = 1/12 OR 1
C
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                   ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      DIMENSION ARRAY(I1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
      IF (JFLAG.EQ.0) GO TO 5
      FD=1./7.
      FM=0.0
      FH=0.0
      GO TO 6
    5 READ 1,ICLASS,FH,FD,FM
    1 FORMAT(I4,4X,3F8.7)
      IF(ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)
      IF (FH+FD+FM.EQ.0.0) GO TO 10
C
C     DETERMINE DEFAULT VALUES
C
      IF (FD.EQ.0.0) FD=1./7.
    6 IF (FM.NE.0.0) GO TO 7
      FM=1./12.
      IF (DAYS.GE.365.) FM=1.
    7 IF (FH.NE.0.0) GO TO 10
      IF(IHR1.GT.6.AND.IHR1.LT.19.AND.IHR2.GT.6.AND.IHR2.LT.19)FH=1./12.
      IF (IHR1.EQ.1.AND.IHR2.EQ.24) FH=1./24.
   10 FRC=FM*FD*FH*(7./DAYS)*(1.0E+6/3.6)
C
   20 DO 100 N=LSRCE,NSRCE
      DO 30 I=1,NPLTS
      ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC
   30 CONTINUE
      IF (NGEOM.EQ.0) GO TO 100
      DO 40 I=1,NGEOM
      ARRAY(I,N)=SORCE(I+2,N)
   40 CONTINUE
      IF (IPT.EQ.1) ARRAY(10,N)=SORCE(2,N)
  100 CONTINUE
      RETURN
      END
```

SUBROUTINE METHD

Purpose:

To calculate diurnal emissions using the temporal distribution arrays for fuel handling activities.

Input:

None

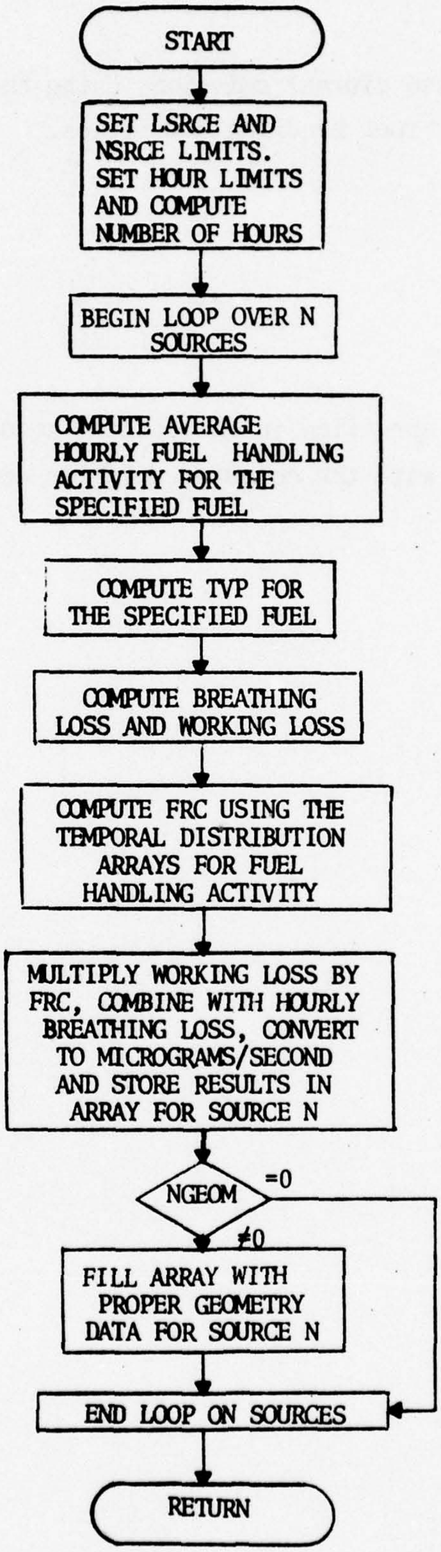
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines Called:

None

SUBROUTINE METHD



```
      SUBROUTINE METHD(MAXN,ARRAY,I1,I2)
C
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE
C     TEMPORAL DISTRIBUTION ARRAYS FOR FUEL HANDLING ACTIVITIES
C
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                   ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON/PERIOD/IMONTH,NODAYS,IDAY,IHR1,IHR2
      COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,
     .  JSTAB,HLID,TEMF,TEMK
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON/MONMET/ TMBAR
      DIMENSION ARRAY(I1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
      NHI=IHR2
      IF(IHR1.GT.IHR2) NHI=24+IHR2
      HRS=NHI-IHR1 + 1
      DO 30 N=LSRCE,NSRCE
      FLHOUR=0.
      IDF=SORCE(14,N)
      DO 10 I=IHR1,NHI
      II=I
      IF(I.GT.24) II=I-24
      FLHOUR=FLHR(II,IDF)+FLHOUR
   10 CONTINUE
      FLHOUR=FLHOUR/HRS
      TVP=EXP(ALPHA(IDF)-BETA(IDF)/(5.*(TMBAR-32.)/9.+273.))
      BRLOSS=SORCE(13,N)*(TVP/(14.7-TVP))**0.69
      WRKLOS=SORCE(12,N)*TVP
      FRC=FLMO(IMONTH,IDF)*FLDY(IDAY,IDF)*FLHOUR*(7./DAYS)
C
      ARRAY(12,N)=(BRLOSS/(365.*24.)+WRKLOS*FRC)*1.E+6/3.6
      IF(IPT.EQ.1) ARRAY(10,N)=SORCE(2,N)
      IF (NGEOM.EQ.0) GO TO 30
      DO 20 I=1,NGEOM
      ARRAY(I,N)=SORCE(I+2,N)
   20 CONTINUE
   30 CONTINUE
      RETURN
      END
```

SUBROUTINE METHE

Purpose:

To calculate diurnal emissions using the temporal distribution arrays for vehicle activities.

Input:

None

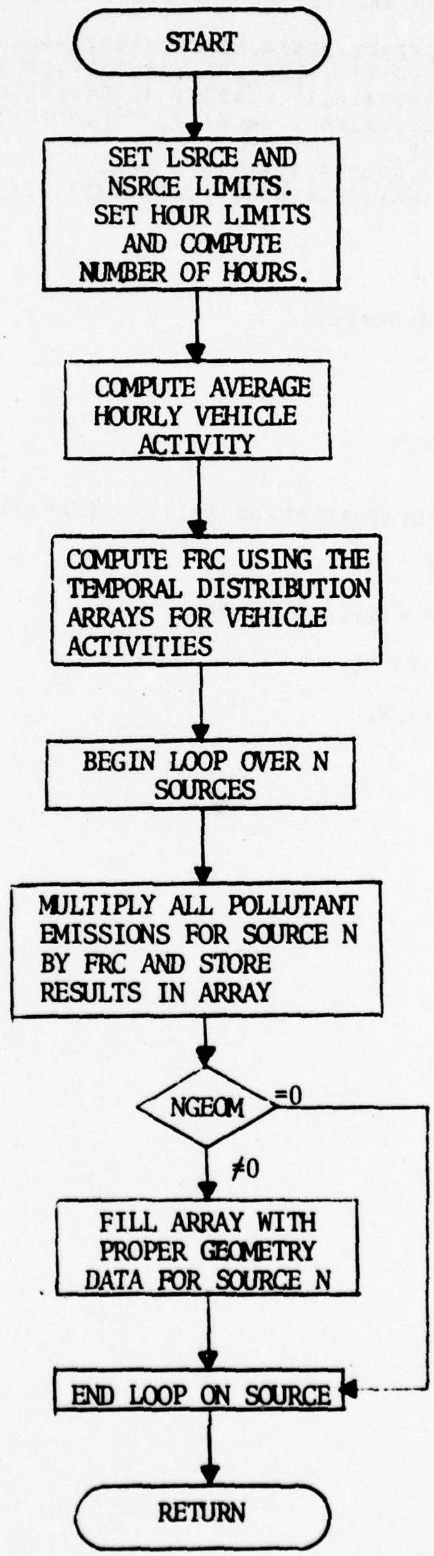
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines Called:

None

SUBROUTINE METHE



```
      SUBROUTINE METHE(MAXN,ARRAY,ARMO,ARDY,ARHR,I1,I2)
C
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE
C     TEMPORAL DISTRIBUTION ARRAYS FOR VEHICLE ACTIVITES
C
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                    ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON/PERIOD/IMONTH,NODAYS,IDAY,IHR1,IHR2
      DIMENSION ARMO(13),ARDY(2),ARHR(24),ARRAY(I1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
      ARHOUR=0.
      NHI=IHR2
      IF(IHR1.GT.IHR2) NHI=24+IHR2
      HRS=NHI-IHR1+1
      DO 10 I=IHR1,NHI
      II=I
      IF(I.GT.24) II=I-24
      ARHOUR=ARHOUR+ARHR(II)
   10 CONTINUE
      ARHOUR=ARHOUR/HRS
      FRC=ARMO(IMONTH)*ARDY(IDAY)*ARHOUR*(7./DAYS)*(1E+6/3.6)
C
      DO 40 N=LSRCE,NSRCE
      DO 20 I=1,NPLTS
      ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC
   20 CONTINUE
      IF (NGEOM.EQ.0) GO TO 40
      DO 30 I=1,NGEOM
   30 ARRAY(I,N)=SORCE(I+2,N)
   40 CONTINUE
      RETURN
      END
```

SUBROUTINE OUTPUT

Purpose:

To print the pollutant concentrations at all receptors for the environ, airbase, aircraft and total combined sources.

Input:

1. Title information.
2. The RECEP and RECDAT arrays containing receptor and concentration data.

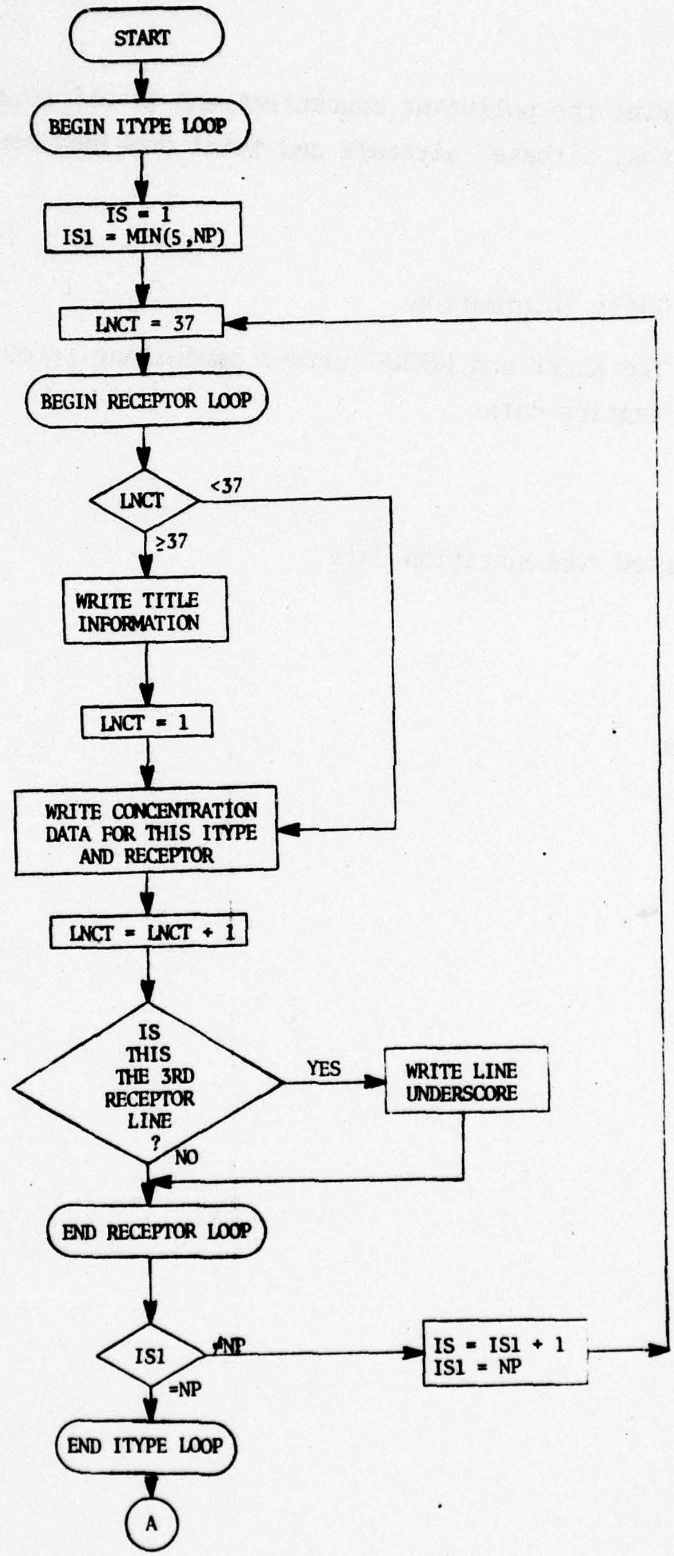
Output:

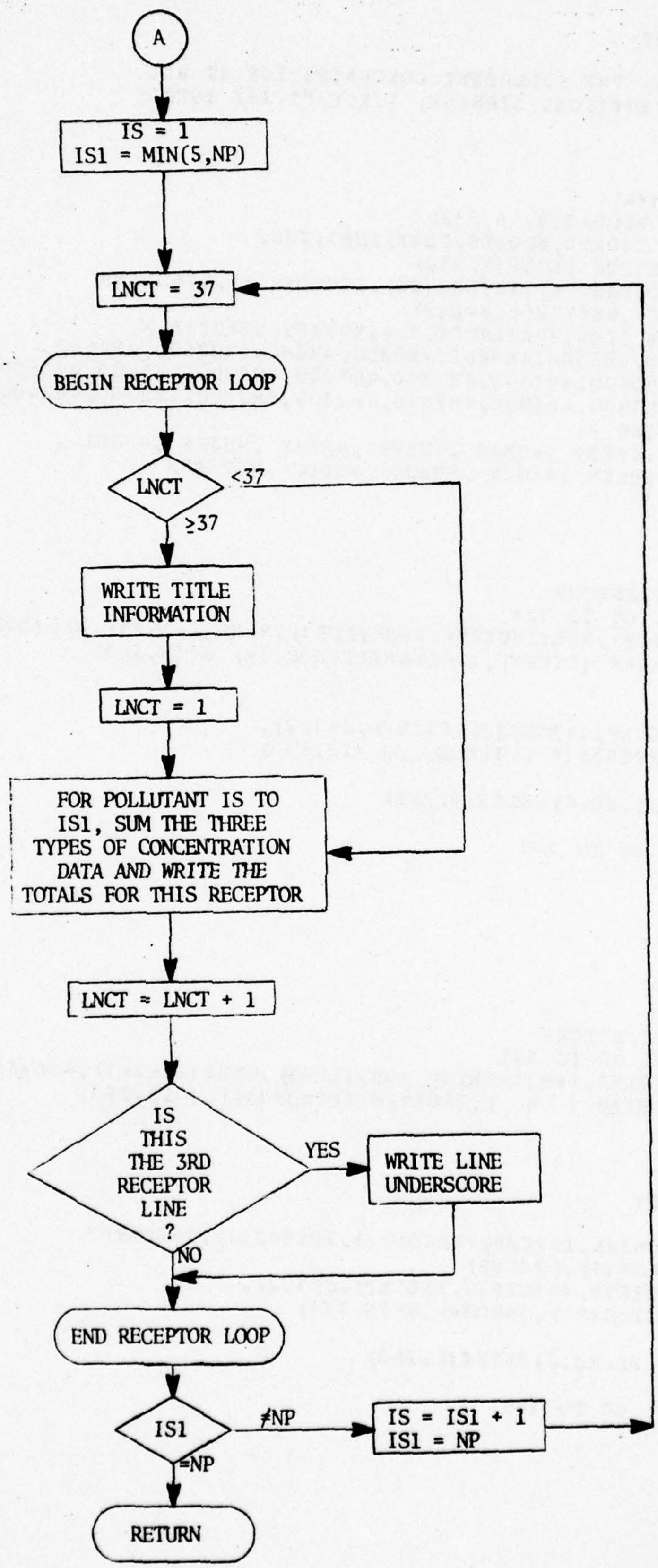
Printed concentration data.

Subroutines Called:

None

SUBROUTINE OUTPUT




A
IS = 1
IS1 = MIN(5,NP)
LNCT = 37
BEGIN RECEPTOR LOOP
LNCT
<37
≥37
WRITE TITLE INFORMATION
LNCT = 1
FOR POLLUTANT IS TO IS1, SUM THE THREE TYPES OF CONCENTRATION DATA AND WRITE THE TOTALS FOR THIS RECEPTOR
LNCT = LNCT + 1
IS THIS THE 3RD RECEPTOR LINE ?
YES
WRITE LINE UNDERSCORE
NO
END RECEPTOR LOOP
IS1
≠NP
IS = IS1 + 1
IS1 = NP
=NP
RETURN

```
      SUBROUTINE OUTPUT
C
C   THIS ROUTINE PRINTS THE POLLUTANT CONCENTRATION AT ALL
C   RECEPTORS FOR THE ENVIRON, AIRBASE, AIRCRAFT AND TOTAL
C   COMBINED SOURCES.
C
      REAL*8 POLNAM
      REAL*8     SORNAM(4)
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /TITL/ POLNAM( 6),TITLE1(20),IPCHOS( 6),NXPOL,NP
      DIMENSION NNM(13),NNHR(25),NND(2)
      DATA SORNAM/7HENVIRON,7HAIRPORT,8HAIRCRAFT,5HTOTAL /
      DATA NNHR/4H0000,4H0100,4H0200,4H0300,4H0400,4H0500,4H0600,
     .4H0700,4H0800,4H0900,4H1000,4H1100,4H1200,4H1300,4H1400,4H1500,
     .4H1600,4H1700,4H1800,4H1900,4H2000,4H2100,4H2200,4H2300,4H2400/,
     . NND /4HDAY ,4HEND /,
     .      NNM/4HJAN ,4HFEB ,4HMAR ,4HAPR ,4HMAY ,4HJUN ,4HJUL ,
     .         4HAUG ,4HSEP ,4HOCT ,4HNOV ,4HDEC ,4HYEAR/
      DO 100 ITYPE=1,3
      IS=1
      IS1=MINO(5,NP)
  110 LNCT=37
      DO 120 IRECEP=1,NRECEP
      IF (LNCT.LT.37) GO TO 121
      WRITE(6,220)TITLE1,NNM(IMONTH),NNHR(IHR1),NNHR(IHR2+1),NND(IDAY)
      WRITE(6,200)SORNAM (ITYPE),(POLNAM(IPCHOS(J)),J=IS,IS1)
      WRITE (6,260)
      LNCT=1
  121 WRITE(6,210)IRECEP,(RECEP(J,IRECEP),J=1,2),
     .(RECDAT(ITYPE,IPCHOS(K ),IRECEP),K =IS,IS1)
      LNCT=LNCT+1
      IF(MOD(IRECEP,3).EQ.0)WRITE(6,260)
  120 CONTINUE
      IF (IS1.EQ.NP) GO TO 100
      IS=IS1+1
      IS1=NP
      GO TO 110
  100 CONTINUE
      IS=1
      IS1=MINO(5,NP)
  125 LNCT=37
      DO 130 IRECEP=1,NRECEP
      IF (LNCT.LT.37) GO TO 133
      WRITE(6,220)TITLE1,NNM(IMONTH),NNHR(IHR1),NNHR(IHR2+1),NND(IDAY)
      WRITE(6,200)SORNAM (   4   ),(POLNAM(IPCHOS(J)),J=IS,IS1)
      WRITE (6,260)
      LNCT=1
  133 CONTINUE
      DO 131 J=IS,IS1
      DO 131 K=2,3
  131 RECDAT(1,IPCHOS(J),IRECEP)=RECDAT(1,IPCHOS(J),IRECEP) +
     . RECDAT(K,IPCHOS(J),IRECEP)
      WRITE(6,210)IRECEP,(RECEP(J,IRECEP),J=1,2),
     . (RECDAT(1,IPCHOS(K ),IRECEP),K=IS,IS1)
      LNCT=LNCT+1
      IF(MOD(IRECEP,3).EQ.0)WRITE(6,260)
  130 CONTINUE
      IF (IS1.EQ.NP) GO TO 140
      IS=IS1+1
      IS1=NP
```

```
      GO TO 125                                                          OUTPT062
140   CONTINUE                                                           OUTPT063
200   FORMAT(1H0,96(1H-)/2H I,22X,33HRECEPTOR CONCENTRATION DATA FROM , OUTPT064
     . A8,8H SOURCES,23X,1HI/2H I,94(1H-),1HI/                           OUTPT065
     . 37H I RECEPTOR I    RECEPTOR LOCATION     I,17X,24HEXPECTED ARITHMETOUTPT066
     .IC MEAN,18X,1HI/13H I   NUMBER  I,23X,1HI,59X,1HI/                 OUTPT067
     . 2H I,10(1H-),1HI,23(1H-),1HI,59(1H-),1HI/                         OUTPT068
     . 2H I,10X,1HI,5X,12H(KILOMETERS),6X,1HI,18X,22H(MICROGRAMS/CU. METOUTPT069
     .ER),19X,1HI,/2H I,10X,1HI,5X,1HX,5X,1HI,5X,1HY,                    OUTPT070
     . 5X,4(3HI    ,A8,1X),3HI    ,A8,2H I)                              OUTPT071
210   FORMAT(2H I,I6,4X,2(1HI,F9.3,2X),1HI,5(1PE10.3,2H I))              OUTPT072
220   FORMAT(1H1,9X,20A4/10H  MONTH = ,A4,12H    PERIOD = ,A4,4H TO ,    OUTPT073
     . A4,16H HOURS ON A WEEK,A4)                                        OUTPT074
260   FORMAT(2HI,10(1H-),1HI,7(11(1H-),1HI))                             OUTPT075
      RETURN                                                             OUTPT076
      END                                                                OUTPT077
```

SUBROUTINE PLRISE

Purpose:

To calculate the effective height and the vertical and horizontal dispersion coefficients for a given stack.

Input:

The stack parameters and current meteorological conditions.

Output:

1. The effective height, h_{eff}.
2. The vertical and horizontal dispersion coefficients, σ_{yo} and σ_{zo}.
3. KSTAB, a flag used in the TRAN function
   = 0, the modified stack height is below the lid
   = 1, the modified stack height is initially above the lid
   = 2, the plume will penetrate the lid.

Procedure:

1. For point sources having no plume rise:

$$h_{eff} = \max (Z_S, H_B, \Delta Z/2.)$$

$$\sigma_{yo} = \Delta Y/2.4$$

$$\sigma_{zo} = \Delta Z/2.4$$

KSTAB = 0 or 1

2. For point sources which may undergo plume rise:
   a. Estimate the wind speed at the top of the aerovane
   b. Modify the stack height by the effect of the stack downwash
   c. Test for building downwash effects. If downwash occurs:

$$h_{eff} = H_B + .5L_B$$

$$\sigma_{yo} = \sigma_{zo} = h_{eff}/1.2$$

   KSTAB = 0 or 1

   d. Test to determine if the buoyant plume rise is significant.

e. Check for an inversion

f. Compute the plume rise using function RISE

g. If no downwash occurs:

$$H_{eff} = Z_S + 2\left(\frac{VS}{U_a} - 1.5\right) \cdot DS + \text{plume rise}$$

$$\sigma_{yo} = \Delta Y/2.4$$

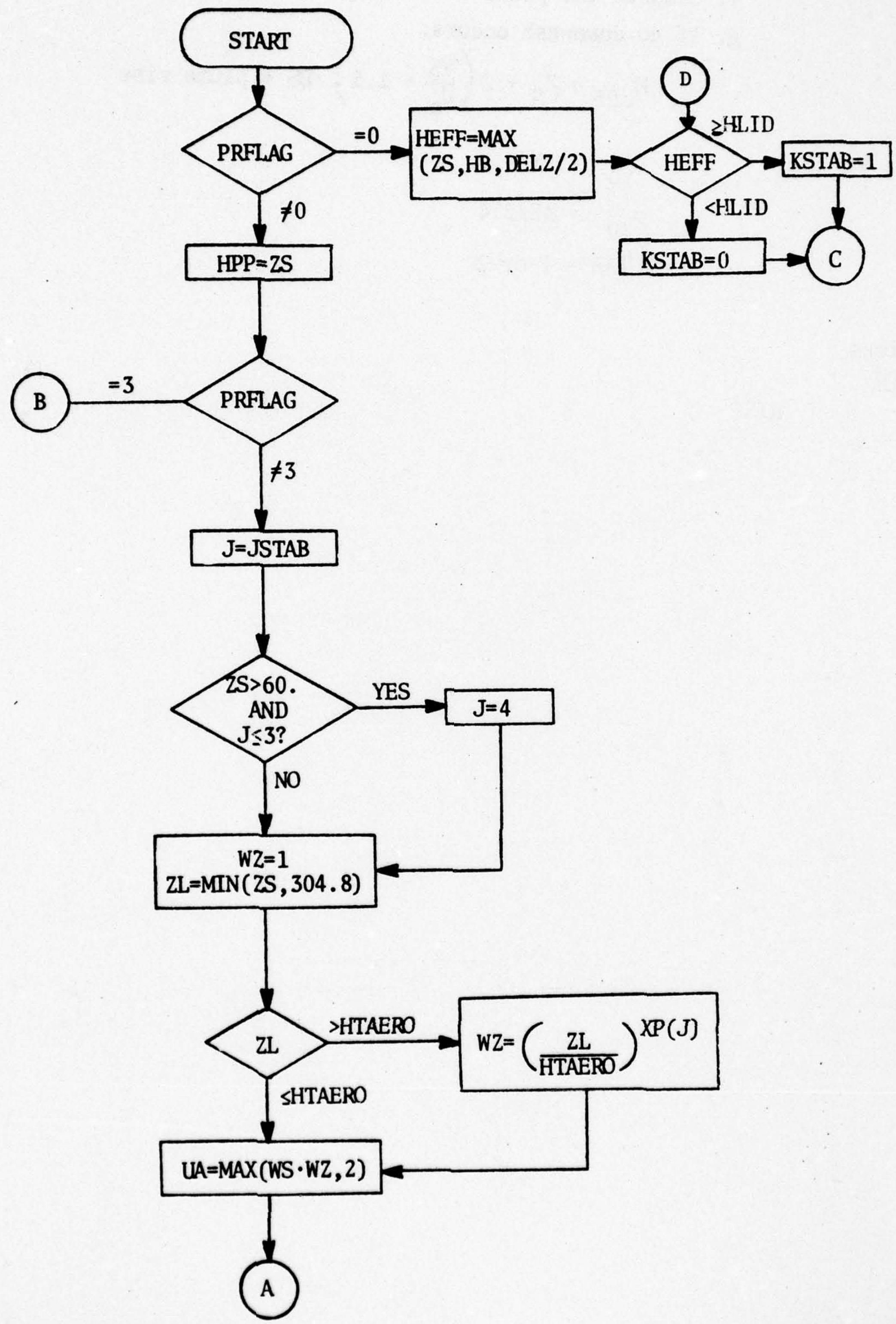
$$\sigma_{zo} = \Delta Z/2.4$$

$$KSTAB = 1 \text{ or } 2$$

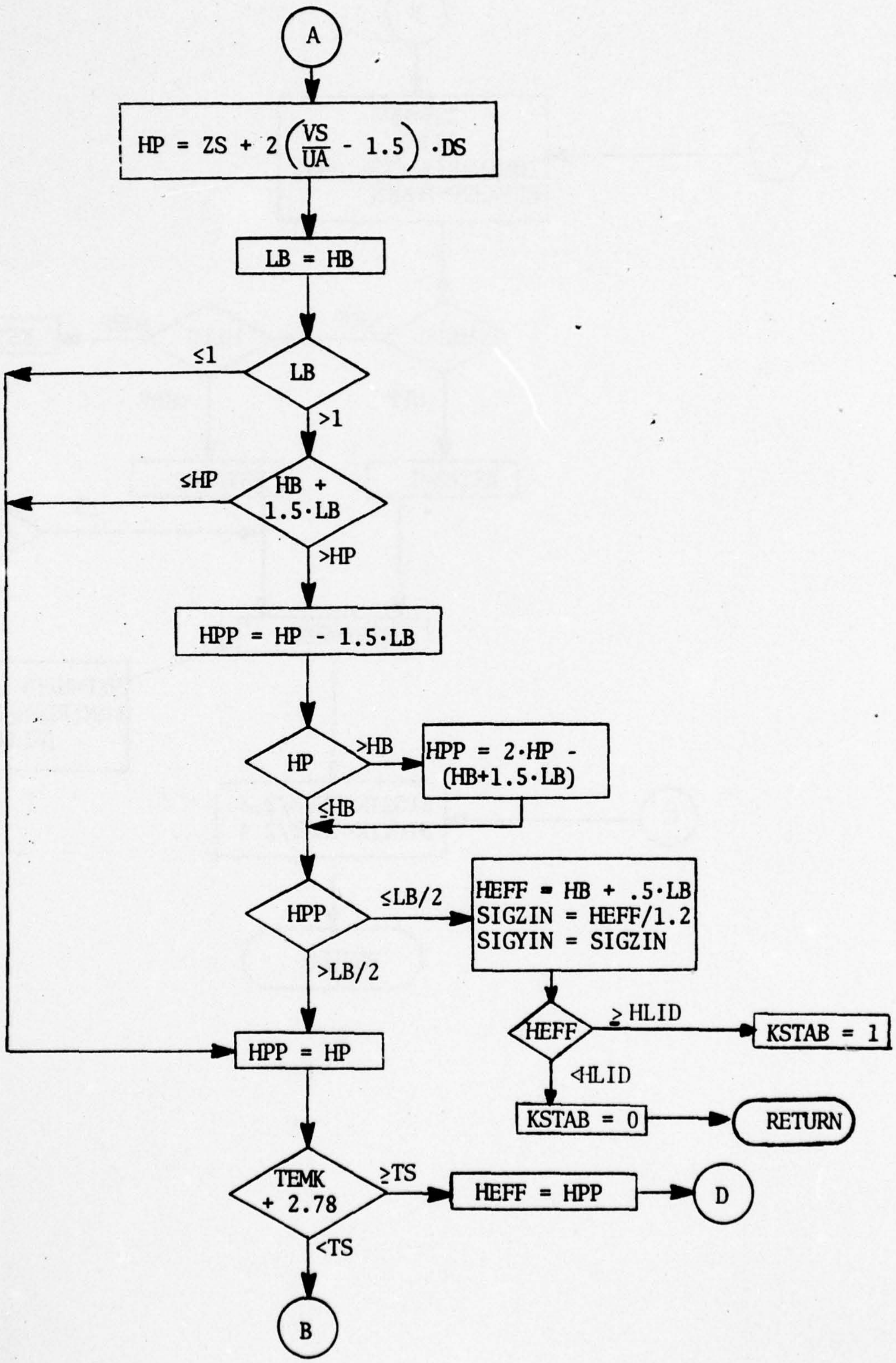
Functions
<u>Called</u>:

RISE

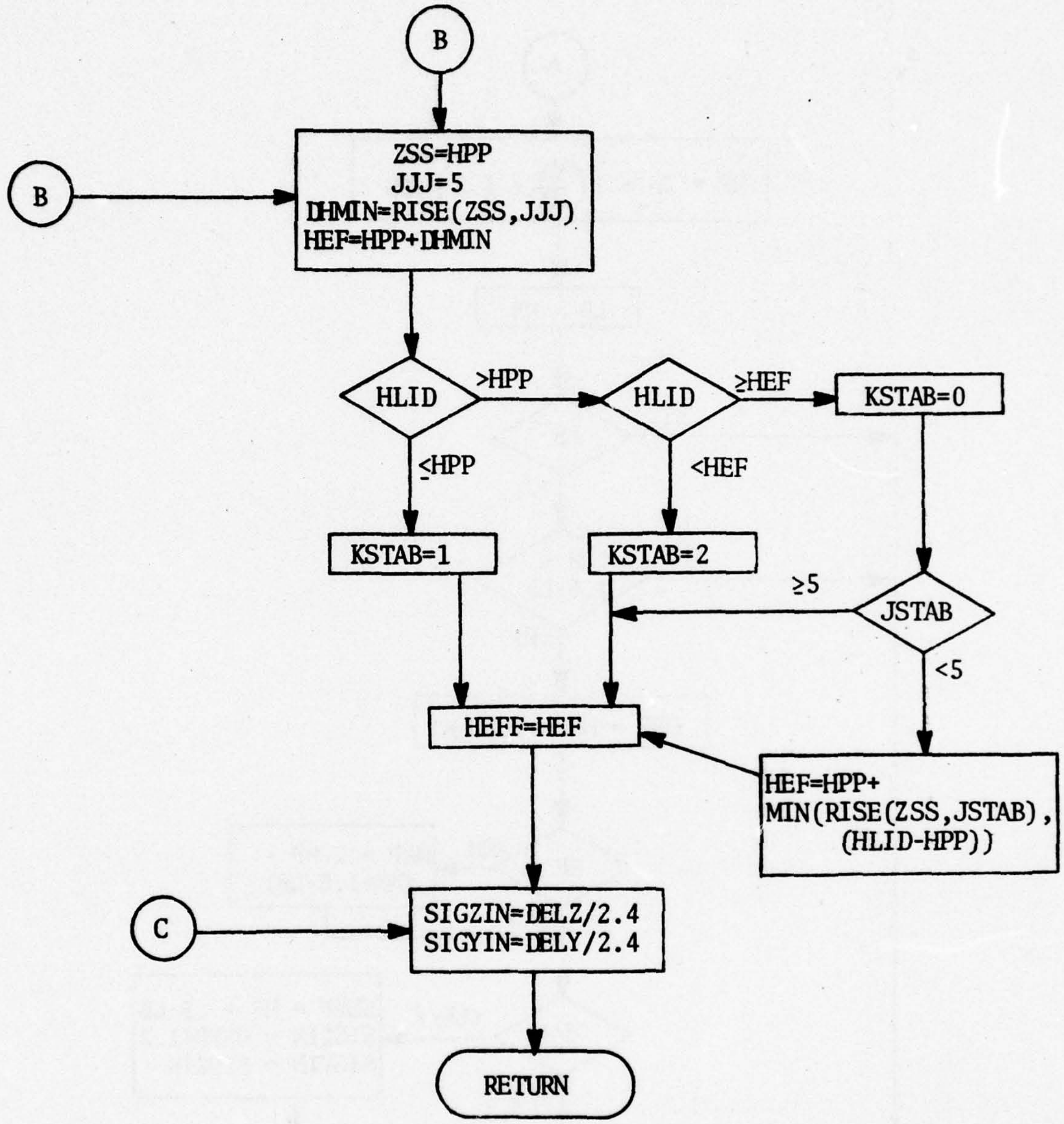
SUBROUTINE PLRISE



SUBROUTINE PLRISE (Cont'd.)



SUBROUTINE PLRISE (Cont'd.)



```
      SUBROUTINE PLRISE(HEFF,KSTAB,SIGZIN,SIGYIN)
C
C   THIS SUBROUTINE CALCULATES THE EFFECTIVE HEIGHT AND THE
C   VERTICAL AND HORIZONTAL DISPERSION COEFFICIENTS
C   FOR A GIVEN STACK
C
      REAL LB
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,
     . TEMK,UA
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,XS,YS,ZS,DELY,DELZ,
     .  TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL
      COMMON /WNDPRO/ XP(6)
      IF (PRFLAG.NE.0) GO TO 100
C
C   FOR AN AREA SOURCE WITH A DIAMETER OF LESS THAN 50 METERS
C   THE EFFECTIVE EMISSION HEIGHT IS SET TO THE MAXIMUM OF
C   Z, THE BUILDING HEIGHT OR DELTA Z/2.0
C
      HEFF=AMAX1(ZS,HB,DELZ/2.)
   50 KSTAB=0
      IF (HEFF.GE.HLID) KSTAB=1
      GO TO 230
C
  100 CONTINUE
      HP=ZS
      IF (PRFLAG.EQ.3) GO TO 130
C
C   FIRST TEST FOR DOWNWASH, THEN COMPUTE PLUME RISE, IF ANY
C
C   FOR TALL STACKS USE STABILITY 4 IN THE WIND PROFILE LAW
C
      J=JSTAB
      IF (ZS.GT.60..AND.J.LE.3) J=4
C
C   COMPUTE THE WINDSPEED AT THE ELEVATION OF THE STACK
C
      WZ=1.0
      ZL=AMIN1(ZS,304.8)
      IF (ZL.GT.HTAERO) WZ= (ZL/HTAERO) **XP(J)
      UA=AMAX1(WS*WZ,2.0)
C
C   COMPUTE STACK DOWNWASH
C
      HP=ZS+2.0*(VS/UA-1.5)*DS
      LB=HB
C
C   BUILDING DOWNWASH TESTS
C
      IF (LB.LE.1.) GO TO 110
      IF (HP.GE.(HB+1.5*LB)) GO TO 110
      HPP=HP-1.5*LB
      IF (HP.GT.HB) HPP=2.0*HP-(HB+1.5*LB)
      IF (HPP.GT.(LB/2.0)) GO TO 110
C
C   BUILDING DOWNWASH OCCURS
C
      HEFF=HB+0.5*LB
      SIGZIN=HEFF/1.2
      SIGYIN=SIGZIN
      KSTAB=0
      IF (HEFF.GE.HLID) KSTAB=1
      RETURN
```

```
C
C   NO BUILDING DOWNWASH, TEST FOR PLUME RISE
C
  110 HPP=HP
      IF (TS.GT.(TEMK+2.78)) GO TO 130
C
C   COLD PLUME
C
      HEFF=HPP
      GO TO 50
  130 CONTINUE
C
C   PLUME RISE EXPECTED TO BE SIGNIFICANT
C   CALCULATE MINIMUM PLUME RISE
C
      ZSS=HPP
      JJJ=5
      DHMIN=RISE(ZSS,JJJ)
      HEF=HPP+DHMIN
C
C   TEST FOR INTERFERENCE OF LID WITH MODIFIED PHYSICAL STACK
C   HEIGHT AND PLUME
C
      IF (HLID.GT.HPP) GO TO 220
C
C   LID INTERFERES WITH STACK HEIGHT, USE STABILITY 5 WITH
C   INFINITE LID HEIGHT
C
      KSTAB=1
      GO TO 225
C
C   LID INTERFERES WITH PLUME, USE STABILITY 5 WITH INFINITE LID
C
  220 IF (HLID.GE.HEF) GO TO 221
      KSTAB=2
      GO TO 225
C
C   CALCULATE PLUME RISE, PLUME CANNOT PENETRATE THE LID
C
  221 KSTAB=0
      IF (JSTAB.LT.5) HEF=HPP+AMIN1(RISE(ZSS,JSTAB),(HLID-HPP))
  225 CONTINUE
      HEFF=HEF
  230 SIGZIN=DELZ/2.4
      SIGYIN=DELY/2.4
      RETURN
      END
```

SUBROUTINE POLSOR

Purpose:

To direct the calls to the proper diffusion routine for all input sources.

Input:

1. Point source data for:
   a. Environs
   b. Airbase
   c. Aircraft
2. Area source data for:
   a. Environs
   b. Airbase
   c. Aircraft
3. Line source data for:
   a. Environs
   b. Airbase
   c. Aircraft

Output:

SORC, a vector which contains data for the current source to be transferred to the diffusion models.

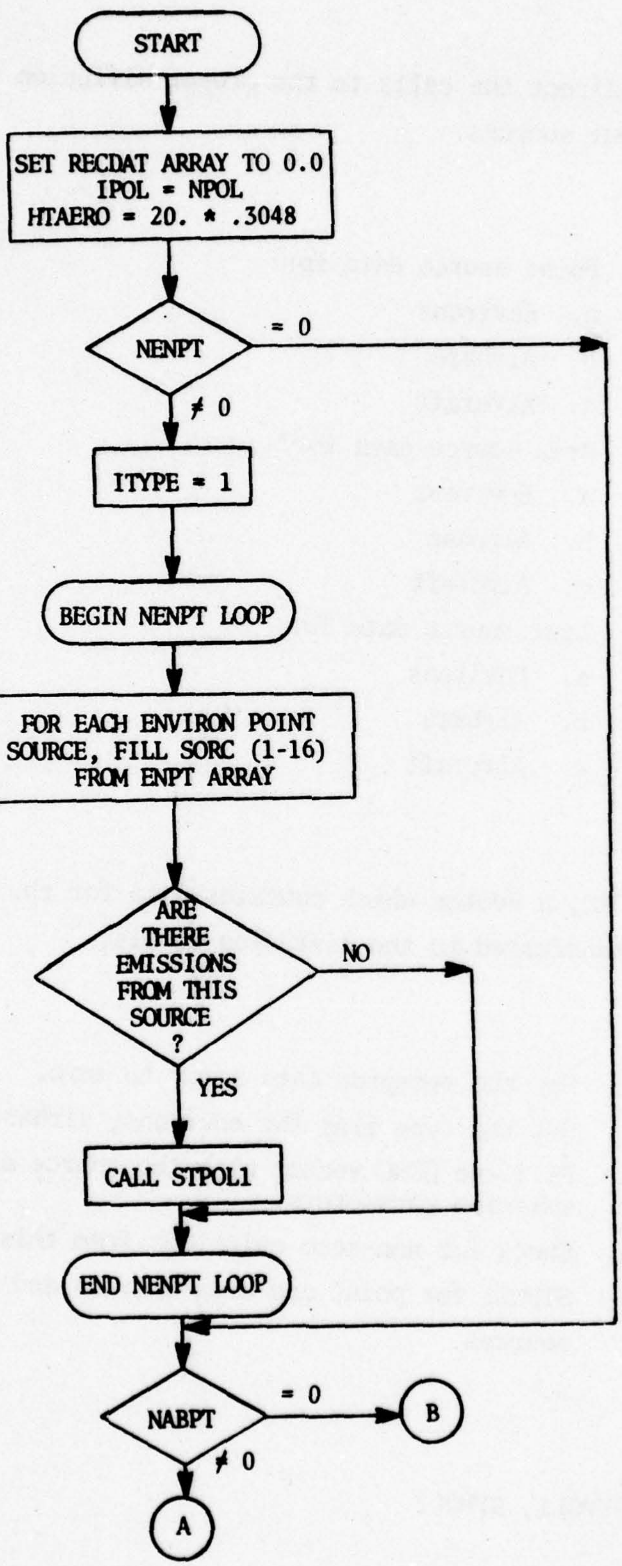
Procedure:

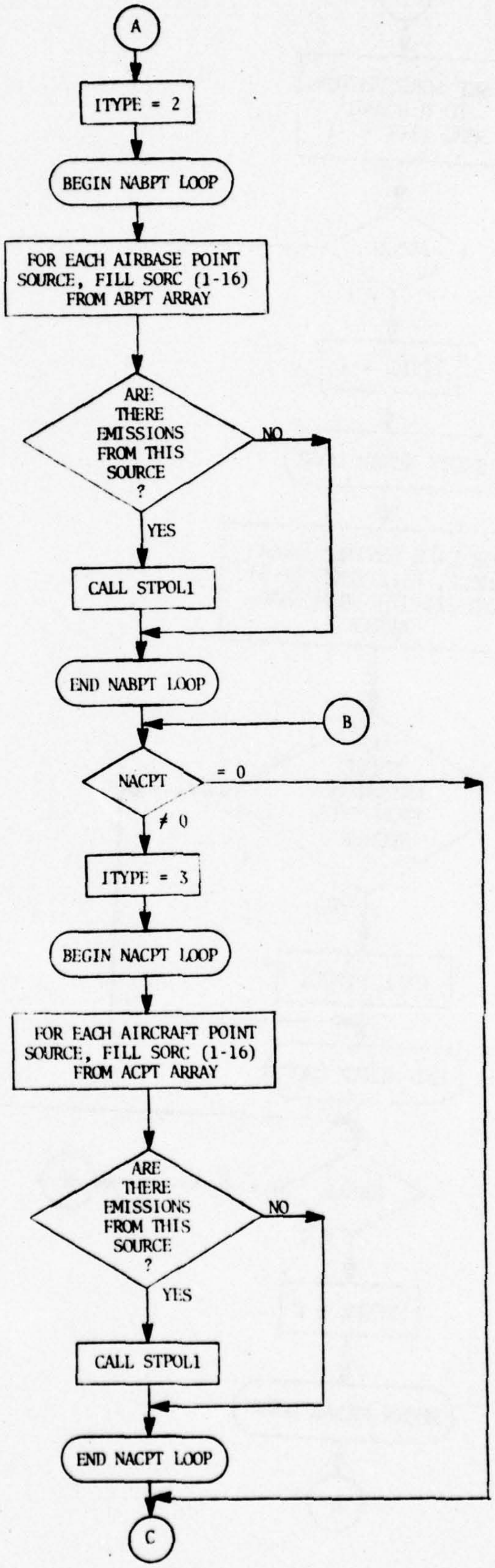
1. Set the receptor data array to zero.
2. Set the type flag for environs, airbase or aircraft.
3. Fill the SORC vector with the source description and emission parameters.
4. Check for non-zero emissions from this source and call STPOL1 for point and area sources and STPOL2 for line sources.

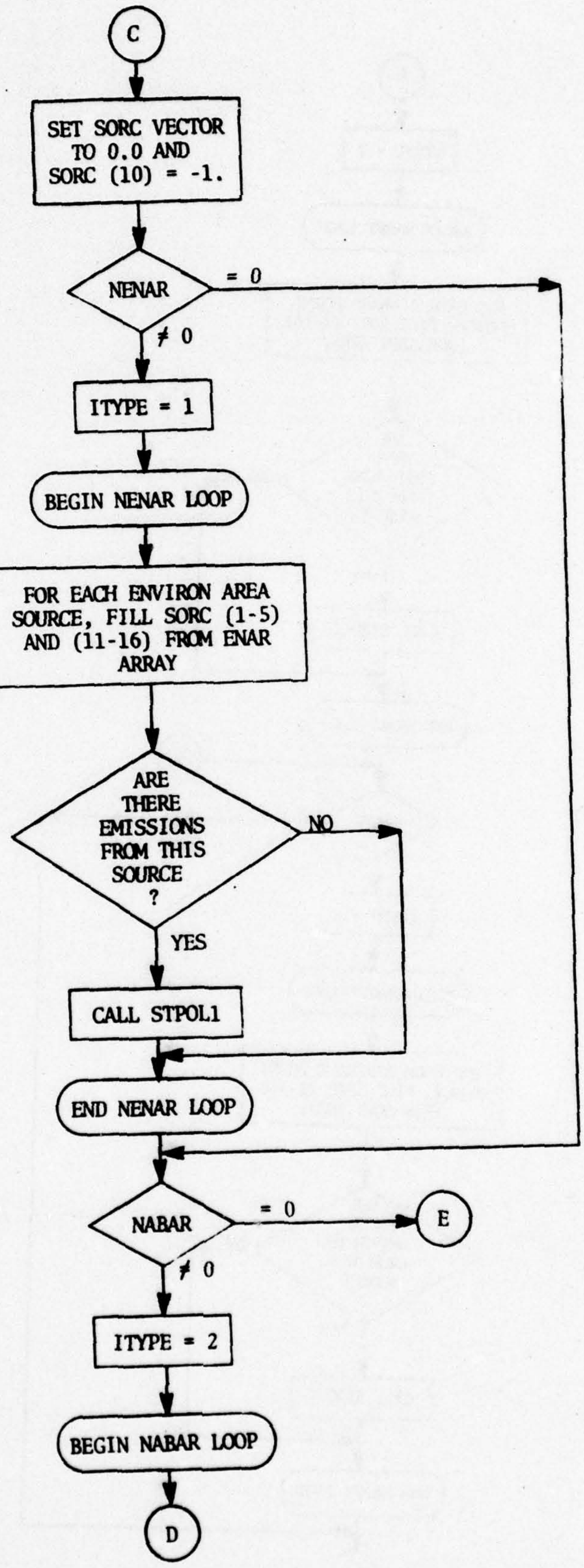
Subroutines Called:

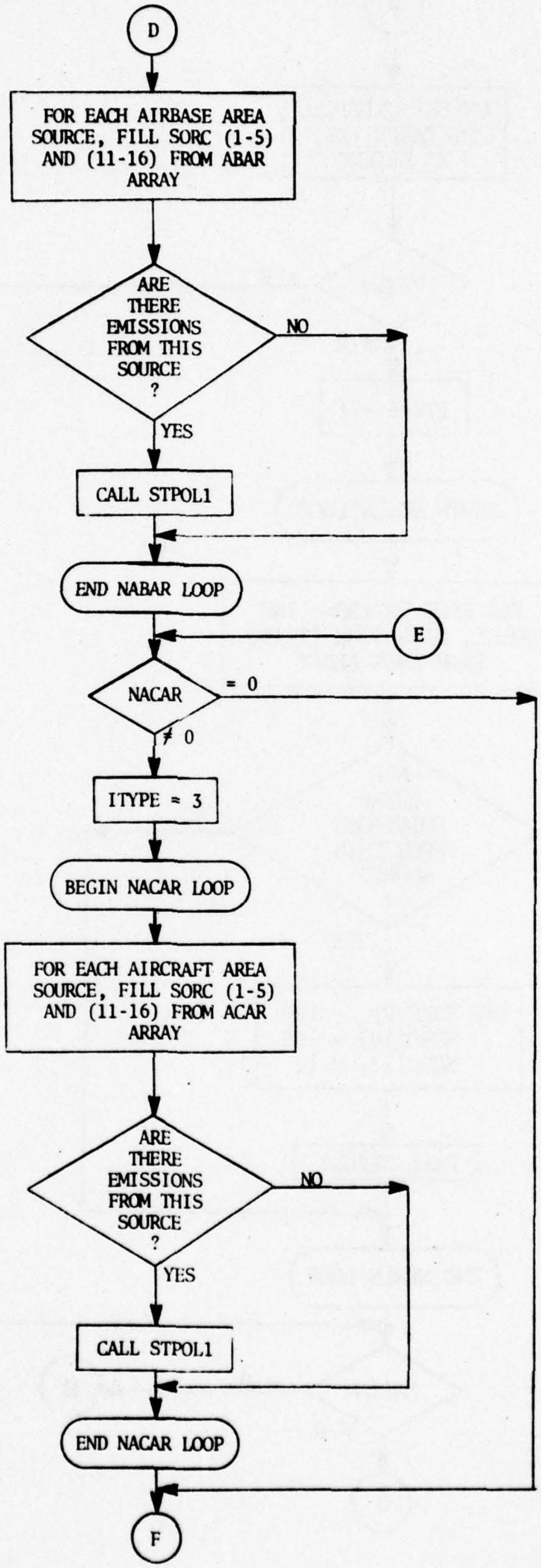
STPOL1, STPOL2

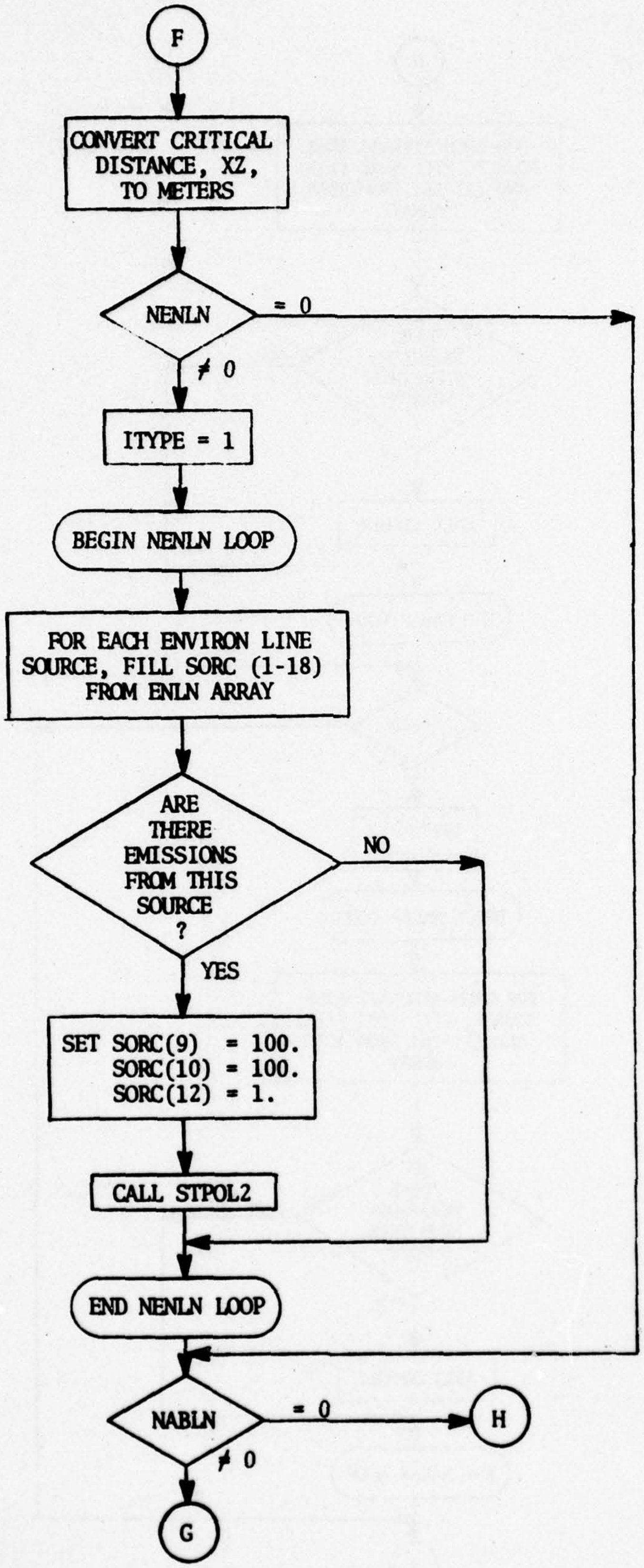
SUBROUTINE POLSOR

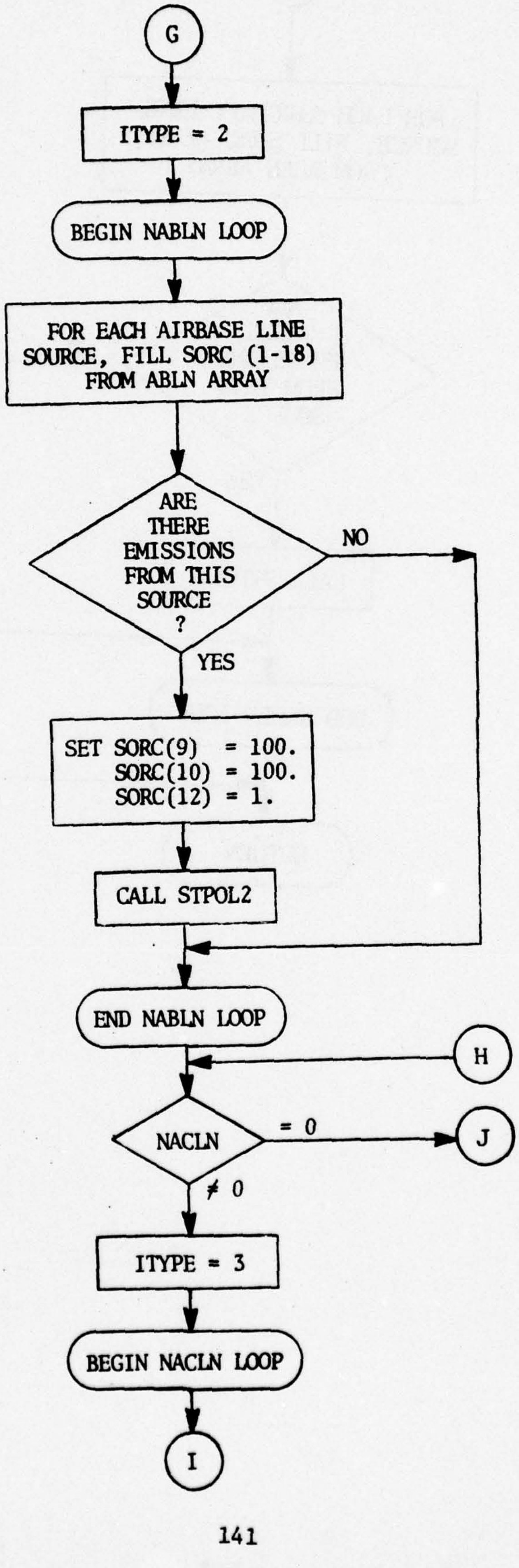


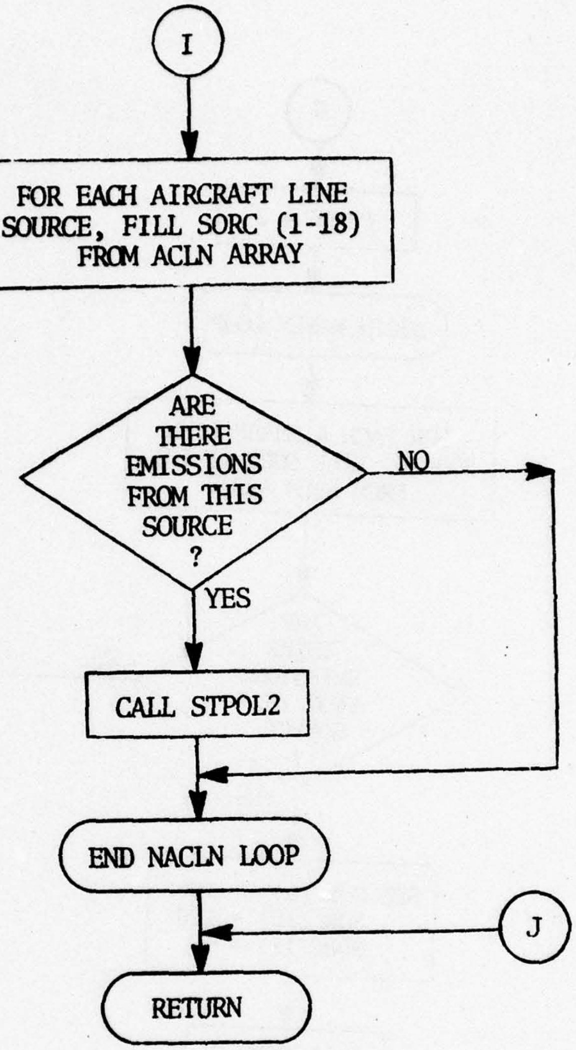

A
ITYPE = 2
BEGIN NABPT LOOP
FOR EACH AIRBASE POINT SOURCE, FILL SORC (1-16) FROM ABPT ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL STPOL1
END NABPT LOOP
B
NACPT
= 0
≠ 0
ITYPE = 3
BEGIN NACPT LOOP
FOR EACH AIRCRAFT POINT SOURCE, FILL SORC (1-16) FROM ACPT ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL STPOL1
END NACPT LOOP
C


C
SET SORC VECTOR TO 0.0 AND SORC (10) = -1.
NENAR
= 0
≠ 0
ITYPE = 1
BEGIN NENAR LOOP
FOR EACH ENVIRON AREA SOURCE, FILL SORC (1-5) AND (11-16) FROM ENAR ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL STPOL1
END NENAR LOOP
NABAR
= 0
E
≠ 0
ITYPE = 2
BEGIN NABAR LOOP
D


D
FOR EACH AIRBASE AREA SOURCE, FILL SORC (1-5) AND (11-16) FROM ABAR ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL STPOL1
END NABAR LOOP
E
NACAR
= 0
≠ 0
ITYPE = 3
BEGIN NACAR LOOP
FOR EACH AIRCRAFT AREA SOURCE, FILL SORC (1-5) AND (11-16) FROM ACAR ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL STPOL1
END NACAR LOOP
F


F
CONVERT CRITICAL DISTANCE, XZ, TO METERS
NENLN
= 0
≠ 0
ITYPE = 1
BEGIN NENLN LOOP
FOR EACH ENVIRON LINE SOURCE, FILL SORC (1-18) FROM ENLN ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
SET SORC(9) = 100.
SORC(10) = 100.
SORC(12) = 1.
CALL STPOL2
END NENLN LOOP
NABLN
= 0
H
≠ 0
G


G
ITYPE = 2
BEGIN NABLN LOOP
FOR EACH AIRBASE LINE SOURCE, FILL SORC (1-18) FROM ABLN ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
SET SORC(9) = 100.
SORC(10) = 100.
SORC(12) = 1.
CALL STPOL2
END NABLN LOOP
H
NACLN
= 0
J
≠ 0
ITYPE = 3
BEGIN NACLN LOOP
I


I
FOR EACH AIRCRAFT LINE SOURCE, FILL SORC (1-18) FROM ACLN ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL STPOL2
END NACLN LOOP
J
RETURN

```
      SUBROUTINE POLSOR
C
C   THIS ROUTINE IS THE DRIVER FOR THE DIFFUSION MODEL.  FOR ALL
C   ENVIRON, AIRBASE AND AIRCRAFT POINTS, AREAS AND LINES,
C   THE SORC VECTOR IS FILLED WITH THE APPROPRIATE SOURCE
C   PARAMETERS AND THEN THE PROPER DIFFUSION ROUTINE IS CALLED
C
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAP(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE     ,HTAERO,SORC(18),IPOL
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /XTRAN/ XZ
C
C   SET RECDAT ARRAY TO 0.0
C
      DO 10 I=1,3
      DO 10 J=1,6
      DO 10 K=1,312
   10 RECDAT(I,J,K) = 0.
      IPOL = NPOL
      HTAERO=20.*.3048
C
C   ENVIRON POINTS
C
      IF (NENPT .EQ. 0) GO TO 126
      ITYPE = 1
      DO 125 I=1,NENPT
      DO 124 J=1,16
  124 SORC(J) = ENPT(J,I)
      DO 224 J=11,16
      IF(SORC(J).NE.0.0) GO TO 225
  224 CONTINUE
      GO TO 125
  225 CALL STPOL1
  125 CONTINUE
C
C   AIRBASE POINTS
C
  126 IF (NABPT .EQ. 0) GO TO 136
      ITYPE = 2
      DO 135 I=1,NABPT
      DO 134 J=1,16
  134 SORC(J) = ABPT(J,I)
      DO 234 J=11,16
      IF(SORC(J).NE.0.0) GO TO 235
  234 CONTINUE
      GO TO 135
  235 CALL STPOL1
  135 CONTINUE
C
C   AIRCRAFT POINTS
C
  136 IF(NACPT .EQ. 0) GO TO 146
      ITYPE = 3
      DO 145 I=1,NACPT
      DO 144 J=1,16
  144 SORC(J) = ACPT(J,I)
      DO 244 J=11,16
      IF(SORC(J).NE.0.0) GO TO 245
```

```
  244 CONTINUE
      GO TO 145
  245 CALL STPOL1
  145 CONTINUE
  146 DO 150 I=1,18
  150 SORC(I) = 0.
      SORC(10) = -1.
C
C   ENVIRON AREAS
C
      IF (NENAR .EQ. 0) GO TO 156
      ITYPE = 1
      DO 155 I=1,NENAR
      DO 153 J=1,5
  153 SORC(J) = ENAR(J,I)
      DO 154 J=6,11
  154 SORC(J+5) = ENAR(J,I)
      DO 253 J=11,16
      IF(SORC(J).NE.0.0) GO TO 254
  253 CONTINUE
      GO TO 155
  254 CALL STPOL1
  155 CONTINUE
C
C   AIRBASE AREAS
C
  156 IF (NABAR .EQ. 0) GO TO 166
      ITYPE = 2
      DO 165 I=1,NABAR
      DO 163 J=1,5
  163 SORC(J) = ABAR(J,I)
      DO 164 J=6,11
  164 SORC(J+5) = ABAR(J,I)
      DO 263 J=11,16
      IF(SORC(J).NE.0.0) GO TO 264
  263 CONTINUE
      GO TO 165
  264 CALL STPOL1
  165 CONTINUE
C
C   AIRCRAFT AREAS
C
  166 IF (NACAR .EQ. 0) GO TO 176
      ITYPE = 3
      DO 175 I=1,NACAR
      DO 173 J=1,5
  173 SORC(J) = ACAR(J,I)
      DO 174 J=6,11
  174 SORC(J+5) = ACAR(J,I)
      DO 273 J=11,16
      IF(SORC(J).NE.0.0) GO TO 274
  273 CONTINUE
      GO TO 175
  274 CALL STPOL1
  175 CONTINUE
C
C   CRITICAL DISTANCE, XZ, MUST BE CONVERTED TO METERS FOR LINE MODEL
C
  176 XZ = XZ  * 1000.
C
C   ENVIRON LINES
C
```

```
      IF (NENLN .EQ. 0) GO TO 186
      ITYPE = 1
      DO 185 I=1,NENLN
      DO 184 J=1,8
  184 SORC(J) = ENLN(J,I)
      DO 384 J=13,18
  384 SORC(J)=ENLN(J-4,I)
      DO 284 J=13,18
      IF(SORC(J).NE.0.0) GO TO 285
  284 CONTINUE
      GO TO 185
  285 SORC( 9) = 100.
      SORC(10) = 100.
      SORC(12) = 1.
      CALL STPOL2
  185 CONTINUE
C
C   AIRBASE LINES
C
  186 IF(NABLN .EQ. 0) GO TO 196
      ITYPE = 2
      DO 195 I=1,NABLN
      DO 194 J=1,8
  194 SORC(J) = ABLN(J,I)
      DO 394 J=13,18
  394 SORC(J)=ABLN(J-4,I)
      DO 294 J=13,18
      IF(SORC(J).NE.0.0) GO TO 295
  294 CONTINUE
      GO TO 195
  295 SORC( 9) = 100.
      SORC(10) = 100.
      SORC(12) = 1.
      CALL STPOL2
  195 CONTINUE
C
C   AIRCRAFT LINES
C
  196 IF (NACLN .EQ. 0) GO TO 206
      ITYPE = 3
      DO 205 I=1,NACLN
      DO 204 J=1,18
  204 SORC(J) = ACLN(J,I)
      DO 304 J=13,18
      IF(SORC(J).NE.0.0) GO TO 305
  304 CONTINUE
      GO TO 205
  305 CALL STPOL2
  205 CONTINUE
  206 CONTINUE
      RETURN
      END
```

SUBROUTINE PSEUDO

Purpose:

To call the SIGCY and SIGCZ functions to find the virtual distance in meters from the source to the pseudo upwind point source.

Input:

1. Initial dispersions in y and z directions.
2. Wind speed and stability class.

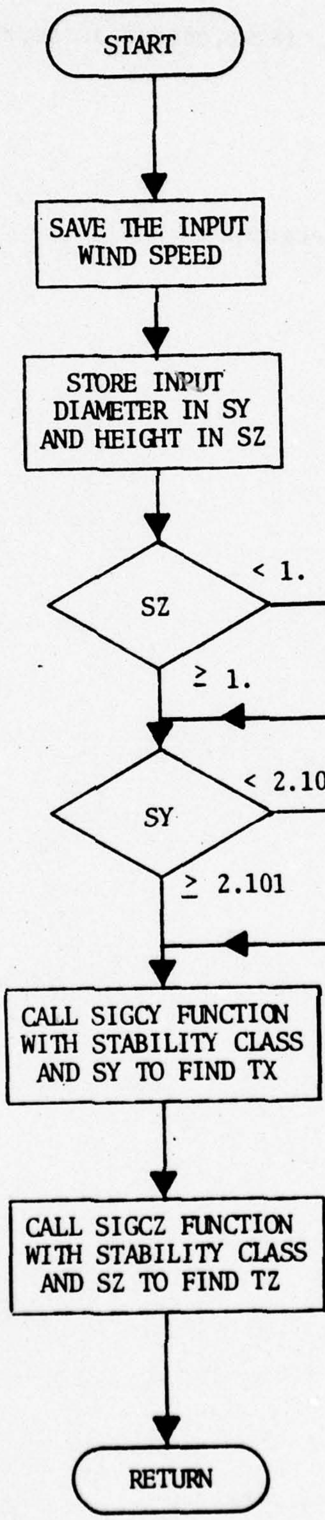
Output:

The virtual distances in the y and z directions.

Functions Called:

SIGCY
SIGCZ

SUBROUTINE PSEUDO



```
      SUBROUTINE PSEUDO(DS,WIN,HS,TY,TZ)
C
C    THIS SUBROUTINE CALLS THE SIGCY AND SIGCZ FUNCTIONS
C    TO FIND THE VIRTUAL DISTANCE FROM THE SOURCE TO THE PSEUDO
C    UPWIND POINT SOURCE
C
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,
     . TEMP,TEMK
      COMMON /WDUM/ WSAVE
C
C    SAVE THE INPUT WIND SPEED
C
      WSAVE=WIN
C
C    SET MINIMUM VALUES FOR INITIAL DISPERSIONS
C
      SY=DS
      SZ=HS
      IF(SZ.LT.1.) SZ=1.0
      IF(SY.LT.2.101) SY=2.101
C
C    FIND DISTANCES IN METERS
C
      TY=SIGCY(JSTAB,SY)
      TZ=SIGCZ(JSTAB,SZ)
      RETURN
      END
```

SUBROUTINE QMOD

Purpose:

To compute the linear distribution, in inverse length, of the pollution along a runway due to aircraft emission during landing or takeoff.

Input:

YSI	Distance along runway measured from tip of exhaust plume near starting end of runway
TAIL	Length or penetration of exhaust plume of aircraft at rest
DL	Length of smoke slug on runway
A	Acceleration (or deceleration) of aircraft
V12	Initial velocity squared
VS	Average velocity of exhaust particles relative to air mass in exhaust plume
WS2	Wind speed squared
WSC	2·wind speed·(- cosine of angle between runway and wind vector)
RR	A/G, where A is acceleration and G is the normalization constant for line density

Output:

QL The linear distribution of pollution

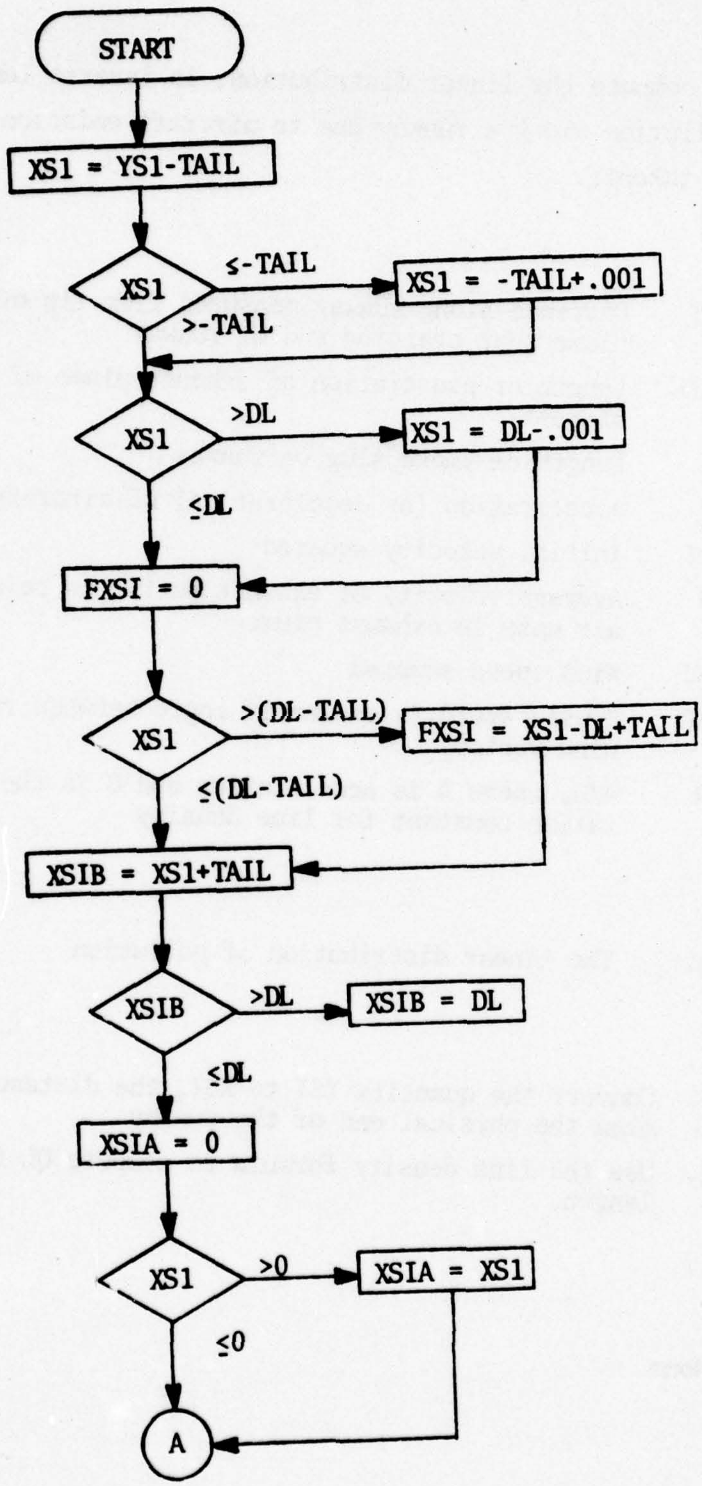
Procedure:

1. Convert the quantity YSI to XSI, the distance measured from the physical end of the runway.
2. Use the line density formula to compute QL in inverse length.

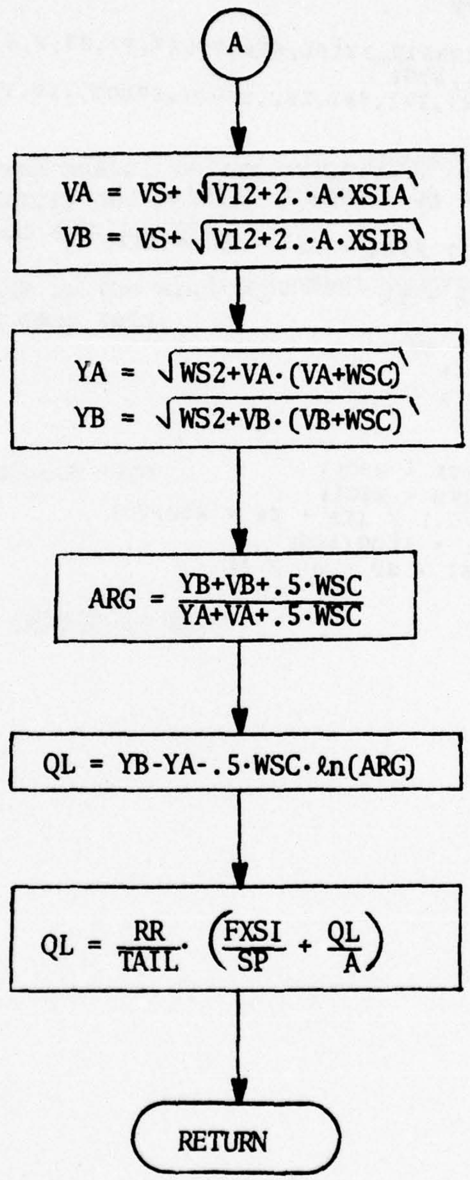
Subroutines Called:

None

SUBROUTINE QMOD



SUBROUTINE QMOD (Contd.)



```
      SUBROUTINE QMOD (YS1,QL)
C
C    THIS ROUTINE COMPUTES THE LINEAR DISTRIBUTION, IN INVERSE LENGTH,
C    OF THE POLLUSION ALONG A RUNWAY DUE TO AIRCRAFT EMMISION
C    DURING LANDING OR TAKEOFF
C
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,
     . V1,V2,DL,TIME,EMIS(6),NPOL
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,
     . WS2,WSC,RR,SP
      XS1 = YS1 - TAIL
      IF (XS1 .LE. -TAIL) XS1 = -TAIL + .001
      IF (XS1 .GT. DL) XS1 = DL - .001
      FXSI = 0.
      IF (XS1 .GT. (DL-TAIL)) FXSI = XS1 - DL + TAIL
   30 XSIB = XS1 + TAIL
      IF (XSIB .GT. DL) XSIB = DL
      XSIA = 0.
      IF (XS1 .GT. 0) XSIA = XS1
      ROOTB = V12 + 2.*A*XSIB
      ROOTA = V12 + 2.*A*XSIA
      VA = SQRT(ROOTA) + VS
      VB = SQRT(ROOTB) + VS
      YA = SQRT(WS2 + VA *(VA + WSC))
      YB = SQRT(WS2 + VB *(VB + WSC))
      ARG = (YB + VB + WSC/2.) / (YA + VA + WSC/2.)
      QL = YB - YA - WSC/2. * ALOG(ARG)
      QL = RR / TAIL * (FXSI / SP + QL / A)
      RETURN
      END
```

SUBROUTINE READ

Purpose:

1. To read master source tape, thereby providing the emission inventory and related data to the source emission distribution subroutines.

2. To set up the wind-dependent sources as random access disk data sets.

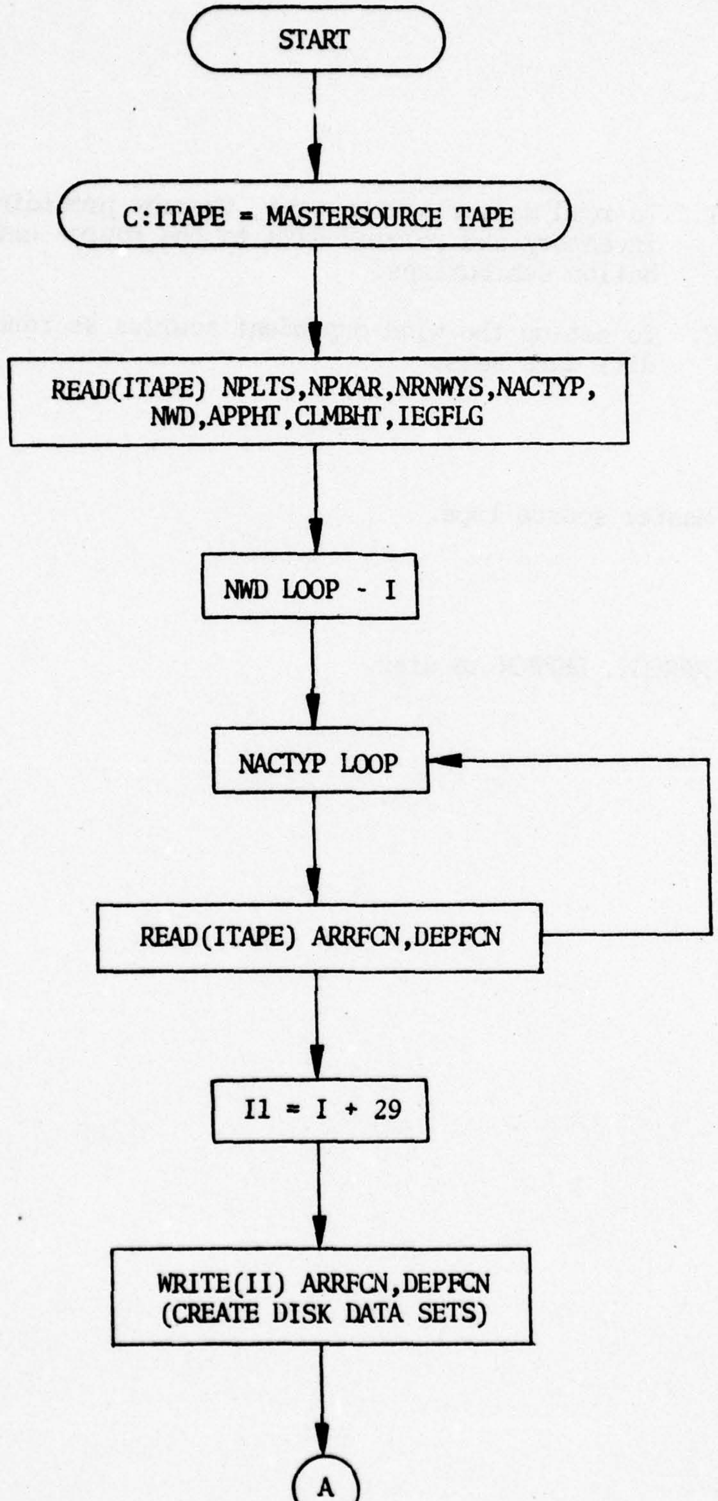
Input:

Master source tape.

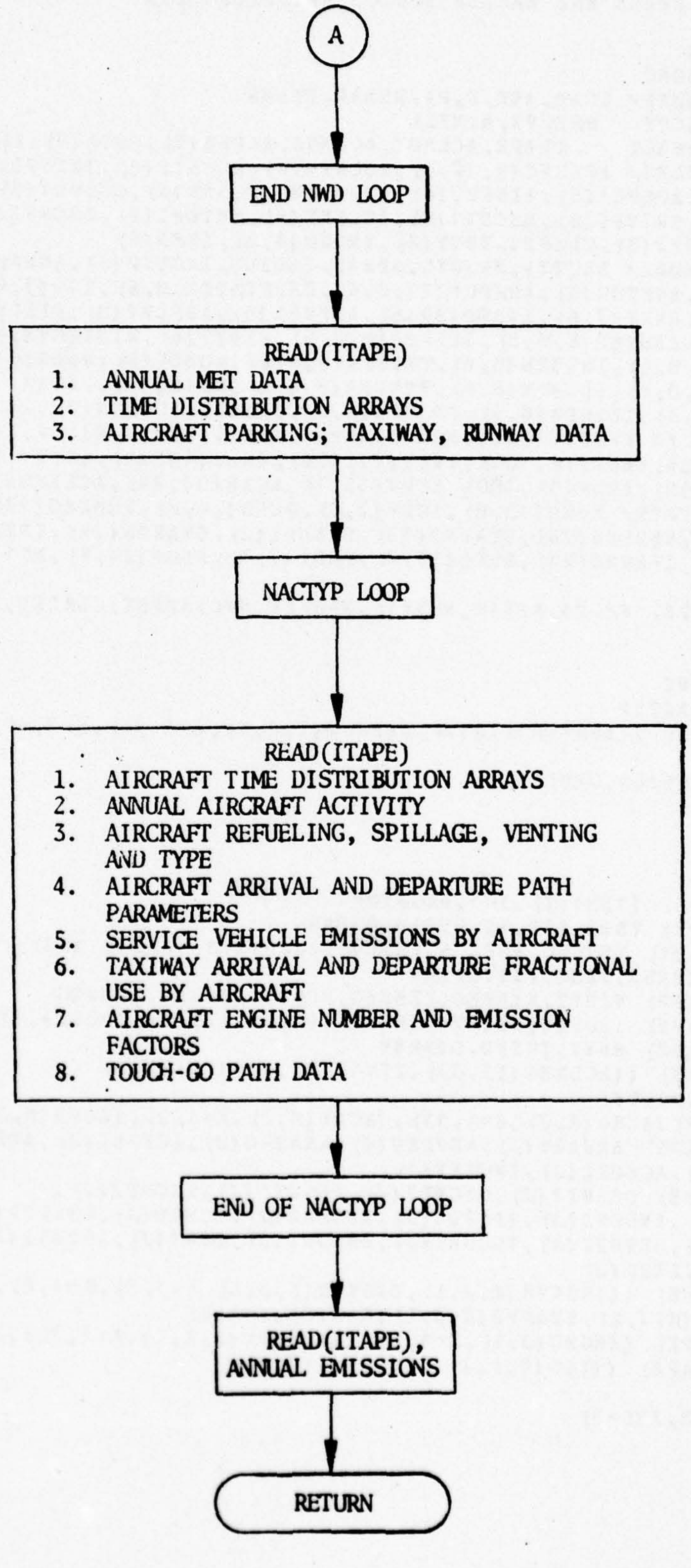
Output:

ARRFCN, DEPFCN to disk.

SUBROUTINE READ



SUBROUTINE READ (Continued)



```
      SUBROUTINE READ
C
C    THIS ROUTINE READS THE MASTER SOURCE EMISSION TAPE
C
      REAL LNDSPD
      INTEGER  ENGNO
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR
      COMMON / RECPT / MRECPT,MAXFIL
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FIDENS(7)
      COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),
     . LNDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),
     . COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2),IDRR(8)
      COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),
     . ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),
     . IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),
     . IOBSEG(16,8,6),IDOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
C
      READ  (ITAPE) NPLTS,NPKAR,NRNWYS,NACTYP,NWD,APPHT,CLMBHT,IEGFLG
     . , NLSEGS
      REWIND 30
      DO 2 I=1,NWD
      DO 5 J=1,NACTYP
      READ (ITAPE) ((ARRFCN(L,J,K),DEPFCN(L,J,K),L=1,24),K=1,6)
    5 CONTINUE
      WRITE(30)ARRFCN,DEPFCN
    2 CONTINUE
      REWIND 30
      MRECPT=1
      MAXFIL=NWD
      READ (ITAPE) (JES1(I),I=1,NACTYP)
      READ (ITAPE) TBAR,ADD,PA,WSBAR,DTBAR
      READ  (ITAPE) VHMLMO,VHML ≤,VHMLqrLceaBmOLceaAAh,c aA¢rLc enM LT
     . CVENDY,CVENHR,FLMO,FLDY,FLHR
      READ  (ITAPE) NIBTT,NIBSEG,IIBSEG,NOBTT,NOBSEG,IOBSEG
      READ  (ITAPE) IDOBTW,IDIBTW,IDPRKA,PAREA,IDIBPA,IDOBPA,NPASQ
      READ  (ITAPE) RNWY,IUSWD,DISRNW
      READ (ITAPE) ((ACLNSG(II,JJ),II=1,12),JJ=1,NLSEGS)
      DO 40 J=1,NACTYP
      READ(ITAPE) (ACMO(K,J),K=1,13),(ACDY(K,J),K=1,2),(ACHR(K,J),K=1,24)
      READ  (ITAPE) ANNARR(J),ANNDEP(J),ANNTGO(J),ACFUEL(J),ARFLVT(J),
     . DPFLVT(J),ACSPIL(J),IACTYP(J)
      READ (ITAPE) DSCNT1(J),DSCNT2(J),ASCNT1(J),ASCNT2(J),
     . TXISPD(J),LNDSPD(J),APSPD1(J),APSPD2(J),TOSPD(J),COSPD1(J),
     . COSPD2(J),SRTUPT(J),EGCHKT(J),SHTDNT(J),TOWT(J),APPHT2(J),
     . COHT1(J),IDRR(J)
      READ (ITAPE) ((ARSVEM(K,J,L),DPSVEM(K,J,L),L=1,5),K=1,6),
     . ((TTARFR(K,J,L),TTDPFR(K,J,L),K=1,8),L=1,6)
      READ (ITAPE) (ENGNO(J,L),L=1,2),((ACEMFC(J,K,L),K=1,10),L=1,6)
      READ  (ITAPE) ((TGO(K,L,J),K=1,3),L=1,4)
   40 CONTINUE
    4 READ(ITAPE,END=3)
      GO TO 4
```

```
3 CONTINUE                                                        READ0062
  RETURN                                                          READ0063
  END                                                             READ0064
```

FUNCTION RISE

Purpose:

To calculate the plume rise using either the Carson-Moses or Holland plume rise formula.

Input:

Stack parameters, current wind speed and stability, temperature, and the plume rise flag.

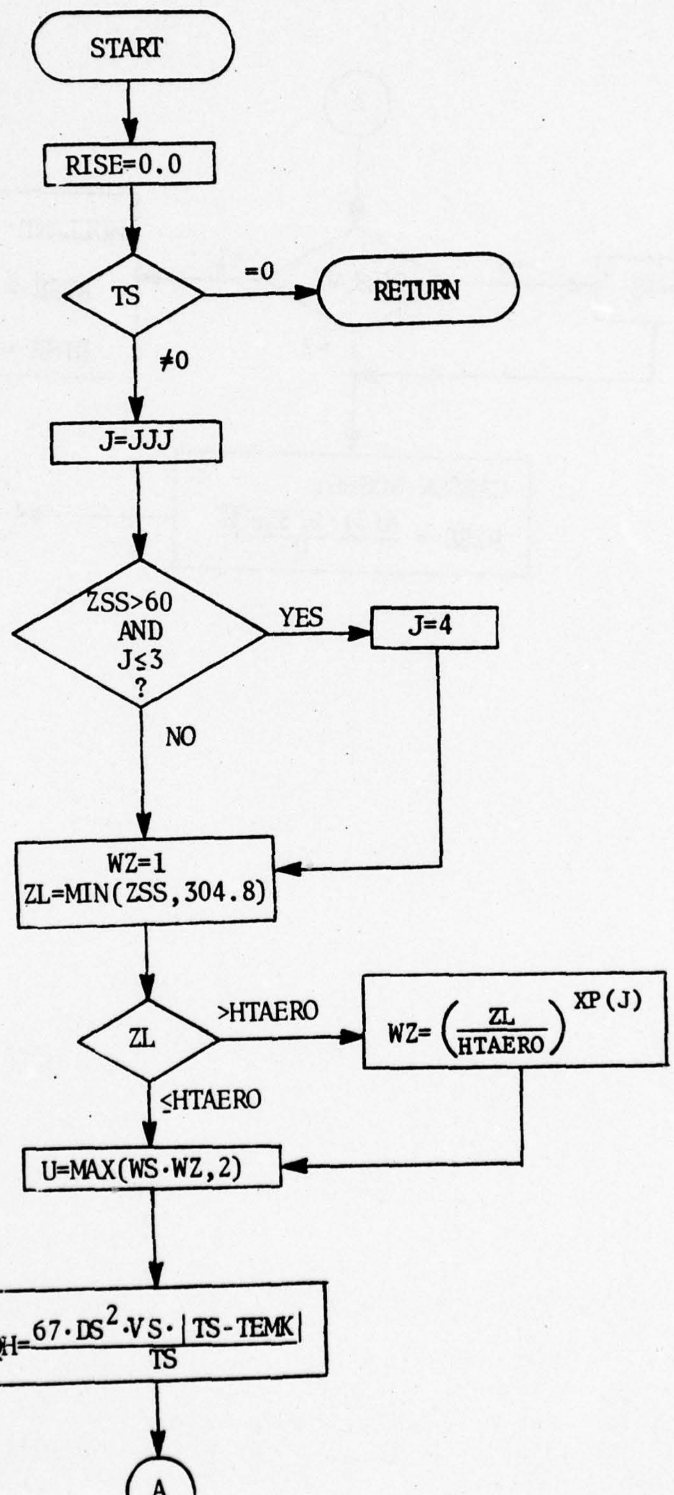
Output:

The height of the plume rise.

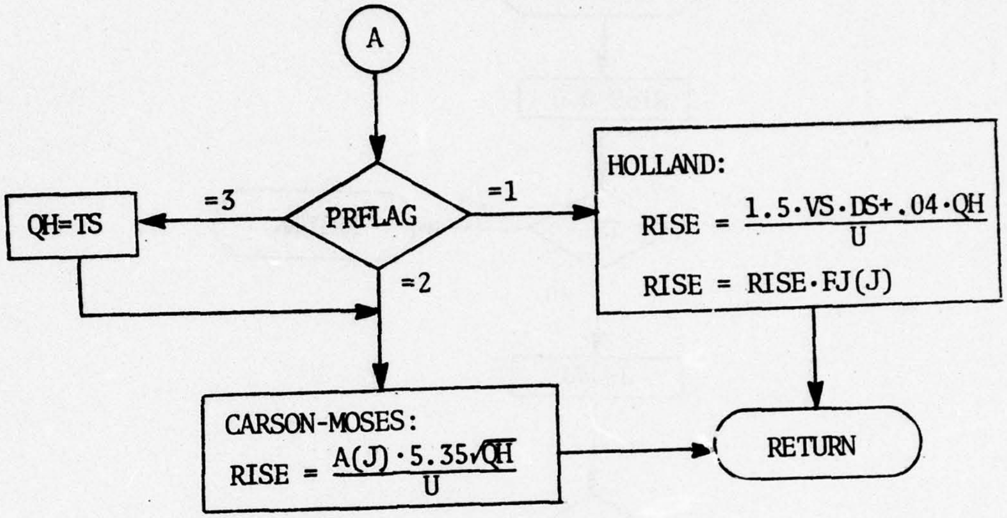
Subroutines Called:

None

FUNCTION RISE



FUNCTION RISE (Cont'd.)



```
      FUNCTION RISE(ZSS,JJJ)
C
C   THIS FUNCTION CALCULATES THE PLUME RISE
C      ZSS IS THE PHYSICAL STACK HEIGHT MODIFIED FOR DOWNWASH
C          EFFECTS, IF ANY
C      JJJ IS THE AMBIENT STABILITY
C
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,XS,YS,ZS,DELY,DELZ,
     . TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL
      DIMENSION A(6),FJ(6)
      COMMON /WNDPRO/ XP(6)
      DATA A /2.65,2.65,2.65,1.08,2*0.68/,
     .     FJ / 1.2,1.2,1.2,1.0,0.8,0.8/
      RISE=0.0
C
C   CHECK THE STACK EXIT GAS TEMPERATURE
C
      IF (TS.EQ.0.0) RETURN
C
C   FOR TALL STACKS USE STABILITY 4 IN THE WIND PROFILE LAW
C
      J=JJJ
      IF(ZSS.GT.60.AND.J.LE.3)J=4
C
C   COMPUTE THE WIND SPEED AT THE ELEVATION OF THE STACK
C   FOR STABILITY J
C
      WZ=1.0
      ZL=AMIN1(ZSS,304.8)
      IF(ZL.GT.HTAERO) WZ=(ZL/HTAERO)**XP(J)
      U=AMAX1(WS*WZ,2.0)
C
C   COMPUTE THE THERMAL EMISSION RATE
C
      QH=67.0*DS*DS*VS*ABS(TS-TEMK)/TS
      IF (PRFLAG.EQ.1.0) GO TO 1
      IF (PRFLAG.EQ.3.0) QH=TS
C
C   CARSON-MOSES PLUME RISE FORMULA
C
      RISE=A(J)*5.35*SQRT(QH)/U
      RETURN
C
C   HOLLAND PLUME RISE FORMULA
C
    1 CONTINUE
      RISE=1.5*VS*DS/U+0.04*QH/U
      RISE=RISE*FJ(J)
      RETURN
      END
```

FUNCTION RRDIST

Purpose:

To calculate the length of runway necessary for takeoff using aircraft dependent equations.

Input:

Aircraft identification, pressure altitude, ambient temperature and wind speed, and aircraft takeoff weight.

Output:

Takeoff length in feet of runway roll to liftoff.

Procedure:

For a given aircraft, use the proper set of takeoff equations provided by the USAF.

Subroutines Called:

None

```
      FUNCTICN RRDIST (IR,PA,T,GW,WS)                                   RRDST000
C                                                                       RRDST001
C     FUNCTION CALCULATES RUNWAY ROLL DISTANCE IN FEET                  RRDST002
C     IR IS AIRCRAFT IDENTIFICATION NUMBER                              RRDST003
C     PA IS PRESSURE ALTITUDE IN HUNDREDS OF FEET                       RRDST004
C     T IS TEMPERATURE IN DEGREES FAHRENHEIT                            RRDST005
C     GW IS AC TAKE OFF WEIGHT IN THOUSAND POUNDS                       RRDST006
C     WS IS THE WIND SPEED IN KNOTS                                     RRDST007
C                                                                       RRDST008
      FGR=0.0                                                           RRDST009
      IF(IR.EQ.100) GO TO 100                                           RRDST010
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,  RRDST011
     123,24,25,26,27,28,29,30,31,32,33,34,35,36,37,100,100,100,100,100, RRDST012
     2 100,100,100,100,100, 100,100,12) ,IR                             RRDST013
    1 CCNTINUE                                                          RRDST014
      GC TO 100                                                         RRDST015
    3 CONTINUE                                                          RRDST016
    2 TOF=-(2.78-8.5714E-4*PA) +(1.82E-2+7.2857E-5*PA) *GW              RRDST017
      GR=(1.184E+1-4.2167E-1*T+1.0E-2*T**2-4.583E-5*T**3) +             RRDST018
     .  (4.194+1.7197E-2*T-9.26018E-4*T**2) *TOF+                       RRDST019
     .  (1.0457+8.40E-3*T+2.117E-4*T**2+2.98E-7*T**3) *TOF**2           RRDST020
      FGR=(GR-(1.15E-1+9.0E-3*GR) *WS) *100.                            RRDST021
      GC TO 100                                                         RRDST022
    4 CCNTINUE                                                          RRDST023
    5 TOF=(1.589+6.883E-3*PA+1.2767E-4*PA**2) +                         RRDST024
     .  (6.819E-3+1.1007E-4*PA-3.924E-7*PA**2) *T+                      RRDST025
     .  (5.979E-5+3.38096E-7*PA+8.532E-9*PA**2) *T**2                   RRDST026
      GR=(-13.25+8.75E-1*GW-1.25E-2*GW**2) +                            RRDST027
     .  (1.3925E+1-9.275E-1*GW+2.125E-2*GW**2) *TOF                     RRDST028
      FGR=(GR-(1.316E-1+8.748E-3*GR) *WS) *100.                         RRDST029
      GC TO 100                                                         RRDST030
    6 TOF=(9.3937E-1+2.0947E-2*PA+2.005E-4*PA**2) +                     RRDST031
     .  (3.746467E-2+4.05625E-4*PA) *T+                                 RRDST032
     .  (1.9928E-4-5.75006E-6*PA+1.40234E-7*PA**2) *T**2                RRDST033
      GR=(1.4307E+1-7.57144E-1*GW+2.6785E-2*GW**2) +                    RRDST034
     .  (1.67257E+1-1.17762*GW+2.7381E-2*GW**2) *TOF                    RRDST035
      FGR=(GR-(2.412799E-2+7.82971E-3*GR) *WS) *100.                    RRDST036
      GO TC 100                                                         RRDST037
    7 TOF=(-1.06E-3+1.674E-2*PA+8.1888E-5*PA**2) +                      RRDST038
     .  (1.36E-2+9.592E-6*PA+1.755E-6*PA**2) *T+                        RRDST039
     .  (5.1099E-5+1.2899E-6*PA-6.123E-9*PA**2) *T**2                   RRDST040
      GR=(-1.423E+1+6.349998E-1*GW+1.6667E-3*GW**2) +                   RRDST041
     .  (6.1857-3.2179E-1*GW+8.214E-3*GW**2) *TOF                       RRDST042
      FGR=(GR-(6.293E-2+7.328E-3*GR) *WS) *100.                         RRDST043
      GO TC 100                                                         RRDST044
    8 TOF=(9.503E-2+3.313E-2*PA+1.3666E-4*PA**2)+                       RRDST045
     .  (2.2546E-2+1.7848E-4*PA-4.04E-6*PA**2) *T+                      RRDST046
     .  (1.3438E-4-1.2166E-6*PA+4.1854E-8*PA**2) *T**2                  RRDST047
      GR=(2.95E+1-2.394*GW+6.497E-2*GW**2) +                            RRDST048
     .  (3.1035+7.52E-2*GW-3.186E-3*GW**2) *TOF+                        RRDST049
     .  (1.2715-1.5535E-1*GW+4.3889E-3*GW**2) *TOF**2                   RRDST050
      FGR=(GR-(-9.0E-2+1.807E-2*GR-7.143E-5*GR**2) *WS) *100.           RRDST051
      GO TO 100                                                         RRDST052
    9 TCF=(3.36455E-3+5.63556E-2*PA) +                                  RRDST053
     .  (4.417E-2-2.031E-3*PA+5.63E-5*PA**2-3.9954E-7*PA**3) *T+        RRDST054
     .  (-9.2E-5+2.08E-5*PA-5.39E-7*PA**2+3.8E-9*PA**3) *T**2           RRDST055
      GR=(1.65838-3.069E-1*GW+8.1363E-2*GW**2)+                         RRDST056
     .  (-3.6111+3.63559E-1*GW) *TOF+                                   RRDST057
     .  (7.3975E-1-8.78749E-2*GW+3.2487E-3*GW**2) *TOF**2               RRDST058
      FGR=(GR-(5.0E-2+7.4E-3*GR) *WS) *100.                             RRDST059
      GC TC 100                                                         RRDST060
   10 TOF=(12.5546-5.7192E-2*PA+1.3075E-4*PA**2)-                       RRDST061
```

```
     .   (2.9032E-2-1.0254E-4*PA-1.45125E-7*PA**2)*T
      GR=((-5.14955E+1+2.57957*GW-1.4425E-2*GW**2)-
     .   (-1.1535E+1+5.915E-1*GW-4.6828E-3*GW**2)*TOF+
     .   (-6.2285E-1+3.2375E-2*GW-2.9056E-4*GW**2)*TOF**2)*1000.
      FGR=(3.305E+1+9.729E-1*GR+2.31E-6*GR**2)-
     .   (8.244+8.3598E-3*GR-1.44E-8*GR**2)*WS
      GO TO 100
   11 TOF=(7.436E-1+4.29E-2*PA)+(2.1276E-2-3.1116E-5*PA)*T
      GR=(1.638E+1-7.78E-1*GW+2.84E-2*GW**2)+
     .   (3.809-1.947E-1*GW+4.264E-3*GW**2)*TOF+
     .   (-1.976E-1+1.5757E-2*GW+4.6189E-4*GW**2)*TOF**2
      FGR=(GR-(8.5E-2+8.25E-3*GR)*WS)*100.
      GO TO 100
   12 TOF=(1.1405-4.659E-3*PA+1.28E-5*PA**2)-
     .   (2.0146E-3-2.46E-5*PA+3.5514E-7*PA**2)*T
      GR=(-3.0029E+1-9.6225E-2*GW+1.25428E-1*GW**2)-
     .   (-7.3845E+1+1.20433*GW+1.7857E-1*GW**2)*TOF+
     .   (-3.57857E+1+7.857E-1*GW+7.14286E-2*GW**2)*TOF**2
      FGR=((3.17413E-1+9.762E-1*GR+2.657E-4*GR**2)-
     .   (1.1114E-1+7.91177E-3*GR+4.40169E-5*GR**2)*WS)*100.
      GO TO 100
   13 TOF=(9.166-5.485E-2*PA)-(3.412E-2-1.8E-4*PA)*T
      GR=(3.02E+2-3.519E+1*GW+1.841*GW**2)-
     .   (1.306E+2-1.277E+1*GW+5.4E-1*GW**2)*TOF+
     .   (2.0687E+1-1.715*GW+6.07E-2*GW**2)*TOF**2-
     .   (1.1578-8.4228E-2*GW+2.46E-3*GW**2)*TOF**3
      FGR=(GR-(9.55E-2+7.15E-3*GR)*WS)*100.
      GO TO 100
   14 TOF=(2.336+1.582E-2*PA+1.172E-4*PA**2)+
     .   (5.604E-3+9.97746E-5*PA-5.8117147E-7*PA**2)*T+
     .   (9.19269E-5-1.34357E-8*PA+1.61411E-8*PA**2)*T**2
      GR=(7.7366-2.52997E-1*GW+2.385E-3*GW**2)+
     .   (-2.1071+4.2586E-2*GW+12.748E-4*GW**2)*TOF
      FGR=(GR-(1.0755E-1+1.4588E-2*GR-7.94156E-5*GR**2)*WS)*100.
      GO TO 100
   15 CONTINUE
      GO TO 100
   16 TOF=(7.6859-1.15E-1*PA+4.413E-4*PA**2)-
     .   (2.925E-2-8.1128E-4*PA+6.999E-6*PA**2)*T-
     .   (2.2289E-4+5.054E-6*PA-7.57E-8*PA**2)*T**2
      GR=(2.546E+1-2.3388*GW+1.0717E-1*GW**2)-
     .   (7.9095-6.7434E-1*GW+2.1045E-2*GW**2)*TOF+
     .   (6.099E-1-5.0858E-2*GW+1.434E-3*GW**2)*TOF**2
      FGR=(GR-(1.16E-1+7.27E-3*GR-3.64E-6*GR**2)*WS)*100.
      GO TO 100
   17 CONTINUE
      GO TO 100
   18 TOF=(2.118+1.058E-2*PA+1.014E-4*PA**2)+
     .   (2.102E-3+1.84E-4*PA-1.177E-6*PA**2)*T+
     .   (1.001E-4-7.046E-7*PA+1.355E-8*PA**2)*T**2
      GR=(1.0E-5)+(-1.9687+4.209E-1*GW+3.9445E-2*GW**2)*TOF
      FGR=(GR-(8.363E-2+1.488E-2*GR-9.78E-5*GR**2)*WS)*100.
      GO TO 100
   19 TOF=(4.65478+6.94444E-3*T)+(3.257E-1+2.7778E-4*T)*(PA/10.)
      GR=(.1457+3.5625E-2*GW-6.763E-5*GW**2)+
     .   (5.1428-3.175E-2*GW+7.0089E-5*GW**2)*TOF
      FGR=(GR-(.1+.0082*GR)*WS)*100.
      GO TO 100
   20 TOF=(1.2192956+2.2091577E-3*PA+3.380102E-4*PA**2)+
     .   (1.4628966E-2+2.6313968E-4*PA-1.3818053E-7*PA**2)*T-
     .   (2.4891E-4-6.875E-6*PA+7.8125E-8*PA**2)*T**2+
     .   (2.20314E-6-6.49E-8*PA+7.47E-10*PA**2)*T**3
```

```
      GR=((2.3806396-5.9265772E-2*GW+6.67969E-4*GW**2)+                 RRDST124
     .   (-1.19933136+5.041098E-2*GW-2.12517E-4*GW**2)*TOF)*10.         RRDST125
      FGR=(1.0+9.7757143E+1*GR+6.4285714E-2*GR**2)-                     RRDST126
     .   (4.8785706+5.4275515E-1*GR+4.438775E-3*GR**2)*WS               RRDST127
      GO TO 100                                                         RRDST128
   21 TOF=(-4.799107E-1 + 3.3165178E-2*PA +2.7902E-4*PA**2) +           RRDST129
     . (2.129E-2 + 2.2538E-4 * PA - 2.9186E-6 * PA ** 2) * T            RRDST130
      GR = (1.16103 + 5.318E-2 * GW + 9.0525E-4 * GW **2 ) +            RRDST131
     . (3.3695E1 - 6.94278E-1 * GW + 3.8559E-3 * GW ** 2 ) * TOF -      RRDST132
     . (-9.041 + 2.307E-1 * GW - 1.264E-3 * GW ** 2 ) * TOF ** 2 +      RRDST133
     . (-1.0708 + 2.477E-2 * GW - 1.108E-4 * GW ** 2 )*TOF ** 3         RRDST134
      FGR=(GR-(2.4131E-1+2.115E-4*GR + 1.935E-4*GR**2)*WS)*100.         RRDST135
      GO TO 100                                                         RRDST136
   22 CONTINUE                                                          RRDST137
   23 TOF=(3.9116E-2+6.3976E-2*PA)+(1.6557E-2-7.6643E-6*PA)*T           RRDST138
      GR=(5.625-9.5E-2*GW+1.3125E-3*GW**2)+                             RRDST139
     .   (8.6496E-1-1.2768E-2*GW+1.077E-4*GW**2)*TOF+                   RRDST140
     .   (4.0067E-1-5.382E-3*GW+3.627E-5*GW**2)*TOF**2                  RRDST141
      FGR=(GR-(1.508E-1+8.625E-3*GR)*WS)*100.                           RRDST142
      GO TO 100                                                         RRDST143
   24 TOF=(5.4067E+1-1.3375E-1*PA-2.2755E-4*PA**2+3.6508E-6*PA**3)-     RRDST144
     .   (7.395E-2-1.71E-4*PA-5.91E-6*PA**2+4.22E-8*PA**3)*T            RRDST145
      GR=(8.6549E+3-7.75196E+1*GW+2.07846E-1*GW**2)-                    RRDST146
     .   (5.6302E+2-4.9948*GW+1.30519E-2*GW**2)*TOF+                    RRDST147
     .   (1.22509E+1-1.07805E-1*GW+2.759985E-4*GW**2)*TOF**2-           RRDST148
     .   (8.8948E-2-7.77463E-4*GW+1.956483E-6*GW**2)*TOF**3             RRDST149
      FGR=(GR-(1.4123219E-1+8.5293578E-3*GR+5.709895E-6*GR**2)*WS)*100. RRDST150
      GO TO 100                                                         RRDST151
   25 TOF=(7.90371+6.68965E-2*PA+2.12622E-4*PA**2)+                     RRDST152
     .   (3.00808E-2+2.67118E-5*PA+9.85E-6*PA**2)*T+                    RRDST153
     .   (1.23149E-4+1.3589E-6*PA-3.1641E-8*PA**2)*T**2                 RRDST154
      GR=(2.1742857+2.04286E-1*GW-1.071429E-2*GW**2)+                   RRDST155
     .   (1.14943-1.2707E-1*GW+5.1785E-3*GW**2)*TOF                     RRDST156
      FGR=(GR-(-2.7327E-2+1.904E-2*GR)*WS+                              RRDST157
     .   (-6.308077E-4+1.94654E-4*GR)*WS**2)*100.                       RRDST158
      GO TO 100                                                         RRDST159
   26 CONTINUE                                                          RRDST160
   27 CONTINUE                                                          RRDST161
   28 CONTINUE                                                          RRDST162
   29 TOF=(7.83935E-1+5.38189E-2*PA)+                                   RRDST163
     .   (1.20408E-2+9.888357E-5*PA-2.32448E-6*PA**2)*T-                RRDST164
     .   (9.72E-6+1.8278E-6*PA-2.405E-8*PA**2)*T**2                     RRDST165
      GR=(3.18978E+1-1.785*GW+3.602E-2*GW**2)+                          RRDST166
     .   (-8.8285+5.1387E-1*GW-5.679E-3*GW**2)*TOF+                     RRDST167
     .   (-1.76441+4.82709E-2*GW)*TOF**2                                RRDST168
      FGR=(GR-(8.6457E-2+1.1414E-2*GR)*WS)*100.                         RRDST169
      GO TO 100                                                         RRDST170
   30 TOF=(-2.890514E-1+5.8370956E-2*PA)+                               RRDST171
     .   (4.161561E-2-3.518445E-5*PA)*T+(-6.0515E-5+3.53095E-6*PA)*T**2 RRDST172
      GR=(-2.684337E+1+3.224954*GW)+(-2.0581519+3.7024356E-1*GW)*TOF+   RRDST173
     .   (-8.861357E-1+8.3093188E-2*GW)*TOF**2                          RRDST174
      FGR=(GR-(1.3583333E-1+9.5833E-3*GR)*WS)*100.                      RRDST175
      GO TO 100                                                         RRDST176
   31 TOF=(7.46275E-1+1.789924E-2*PA+1.667729E-4*PA**2)+                RRDST177
     .   (6.1017875E-3+3.4816947E-4*PA-1.6406229E-6*PA**2)*T+           RRDST178
     .   (1.718525E-4-2.621825E-6*PA+4.184375E-8*PA**2)*T**2            RRDST179
      GR=(-7.2378129E+1+3.8485684E+1*GW-6.565*GW**2+3.916E-1*GW**3)+    RRDST180
     .   (-5.477E+1+2.92E+1*GW-4.975*GW**2+2.906E-1*GW**3)*TOF          RRDST181
      FGR=((-1.607758+1.222176*GR-5.64375E-3*GR**2)-                    RRDST182
     .   (.482382E-1+2.2260152E-2*GR-4.7462116E-4*GR**2)*WS)*100.       RRDST183
      GO TO 100                                                         RRDST184
   32 TOF=(1.996+1.69E-2*PA+2.56E-5*PA**2)+                             RRDST185
```

```
      .  (8.64E-3-7.5E-5*PA+1.61F-6*PA**2) *T
      GR=(6.26E+1-1.299E+1*GW+6.886E-1*GW**2) +
      .  (-1.0004E+2+2.0317E+1*GW-9.67E-1*GW**2) *TOF+
      .  (1.30368E+1-2.689*GW+1.403E-1*GW**2) *TOF**2
      FGR=((-3.3E-1+1.047*GR-8.57E-4*GR**2) -
      .  (4.22E-2+9.47E-3*GR+1.9898E-5*GR) *WS) *100.
      GO TC 100
33    TCF=(6.6742857E-1+4.4226786E-2*PA) +
      .  (1.027143E-2+3.051339E-4*PA) *T+ (1.74994E-4+5.023E-7*PA) *T**2
      GR=(-1.37666666E+1+1.679166666*GW) + (-3.55+4.71875E-1*GW) *TOF
      FGR=(GR-(1.5166666666E-1+1.0083333333E-2*GR) *WS) *100.
      GO TC 100
34    CCNTINUE
35    CCNTINUE
36    TCF=(-9.2083337E-1+5.9113889E-2*PA) + (2.291666E-2-2.7778E-5*PA) *T
      GR=(3.711176E+1-1.640279E+1*GW+2.22809*GW**2) +
      .  (-2.09922E+1+8.6991796*GW-8.4586E-1*GW**2) *TOF+
      .  (2.248949-9.093486E-1*GW+1.061975E-1*GW**2) *TOF**2
      FGR=(GR-(4.3358E-2+2.196E-2*GR) *WS+
      .  (8.79209E-4+8.21219E-5*GR) *WS**2) *100.
      GC TO 100
37    TCF=(-6.46E-1+6.7857E-2*PA+2.723E-4*PA**2) +
      .  (3.69E-2-2.24E-3*PA+3.49E-5*PA**2) *T+
      .  (1.07E-4+3.85E-5*PA-4.688E-7*PA**2) *T**2
      GR=(5.38-1.105*GW+1.14E-1*GW**2) +
      .  (8.02E-1-2.57E-1*GW+2.4F-2*GW**2) *TOF
      FGR=(GR-(1.6E-2+2.44E-2*GR-2.128E-4*GR**2) *WS) *100.
      GC TC 100
100   RRDIST=FGR
      RETURN
      END
```

FUNCTION SIGY
(ENTRY: SIGCY)

Purpose:

To compute the horizontal dispersion coefficient in meters, or at entry SIGCY, to compute the virtual distance corresponding to the initial horizontal dispersion.

Input:

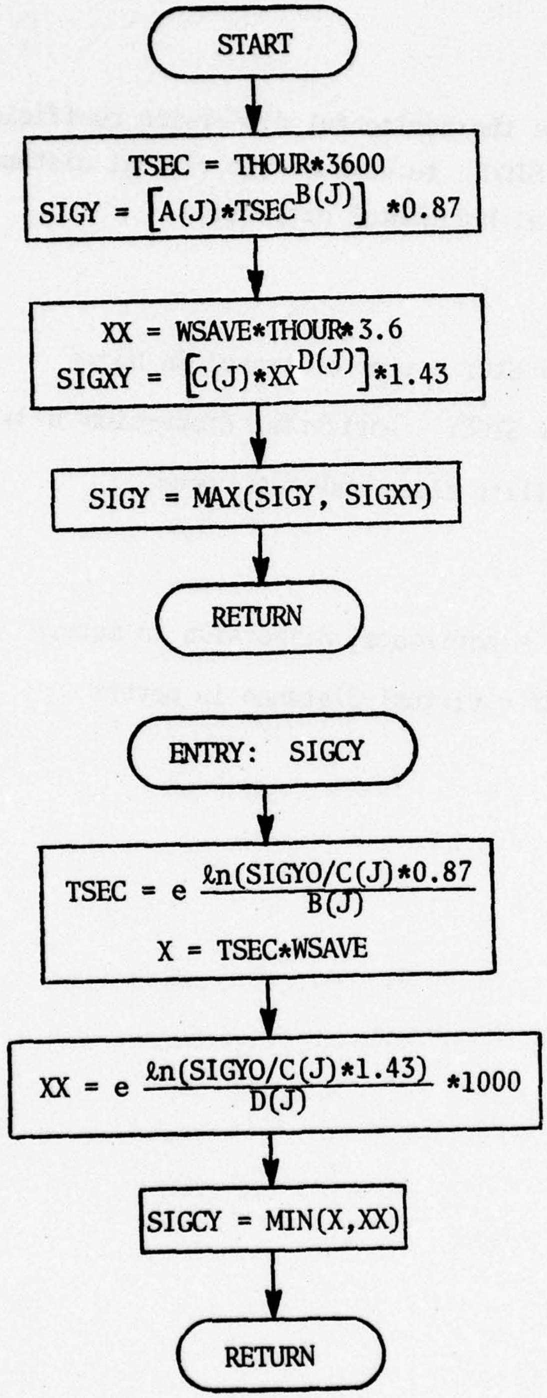
1. Entry SIGY - time of travel in hours
2. Entry SIGCY - horizontal dispersion in meters
3. Stability class and wind speed

Output:

1. SIGY = horizontal dispersion in meters
2. SIGCY = virtual distance in meters

FUNCTION SIGY

ENTRY: SIGCY



```
      FUNCTION SIGY(J,THOUR)
C
C   THIS FUNCTION COMPUTES THE HORIZONTAL DISPERSION COEFFICIENT
C   IN METERS
C
      COMMON /WDUN/ WSAVE
      DIMENSION A(6),B(6),C(6),D(6)
      DATA A/2.1511,1.5454,1.0606,.68465,.59366,.59366/
      DATA B/.87326,.88261,.89031,.88866,.89138,.89138/
      DATA C /212.,155.,100.,68.,50.,34./
      DATA D/0.89,0.91,0.92,0.93,0.90,0.93/
C
      TSEC=THOUR*3600.
      SIGY=(A(J)*TSEC**B(J))*0.87
      XX=WSAVE*THOUR*3.6
      SIGXY=C(J)*(XX**D(J))*1.43
      SIGY=AMAX1(SIGY,SIGXY)
      RETURN
      ENTRY SIGCY(J,SIGY0)
C
C   AT THIS ENTRY THE DISTANCE OR TRAVEL TIME CORRESPONDING TO THE
C   INPUT VALUE OF THE HORIZONTAL DISPERSION IS CALCULATED AND
C   RETURNED AS DISTANCE IN METERS
C
      TSEC=EXP(ALOG(SIGY0/(A(J)*0.87))/B(J))
      X=TSEC*WSAVE
      XX=EXP(ALOG(SIGY0/(C(J)*1.43))/D(J))*1000.
      SIGCY=AMIN1(X,XX)
      RETURN
      END
```

FUNCTION SIGZ

(ENTRY: SIGCZ)

Purpose:

To compute the vertical dispersion coefficient in meters, or at entry SIGCZ, to compute the virtual distance corresponding to the initial vertical dispersion.

Input:

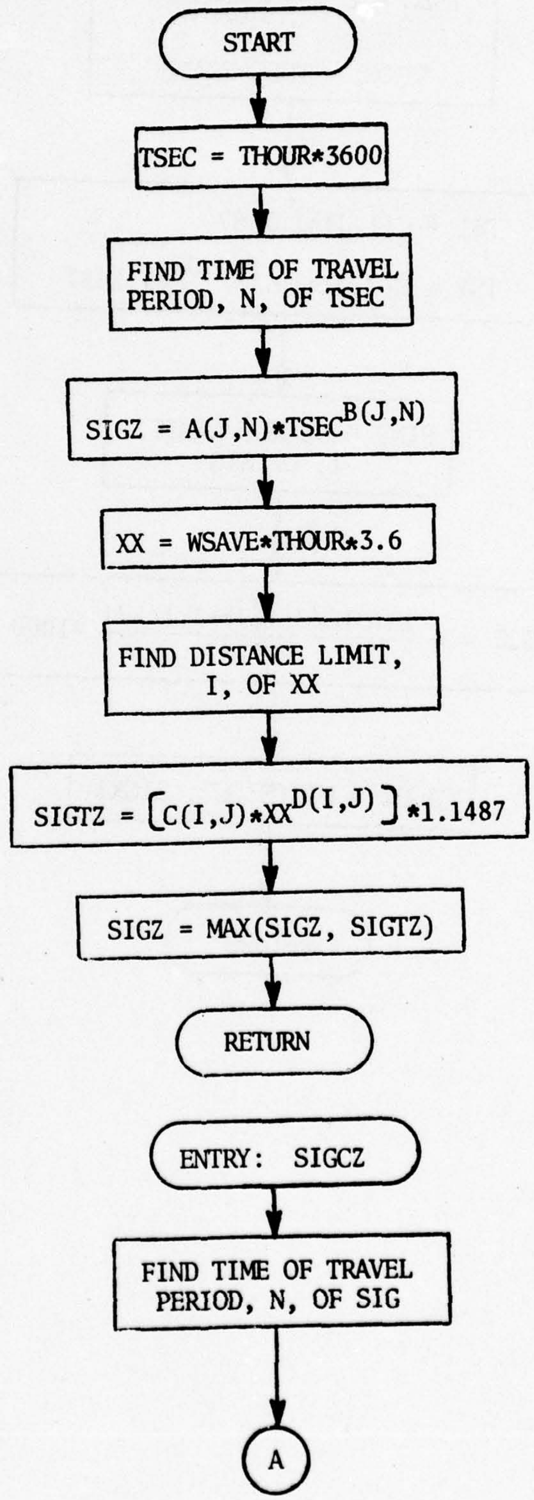
1. Entry SIGZ - time of travel in hours
2. Entry SIGCZ - vertical dispersion in meters
3. Stability class and wind speed

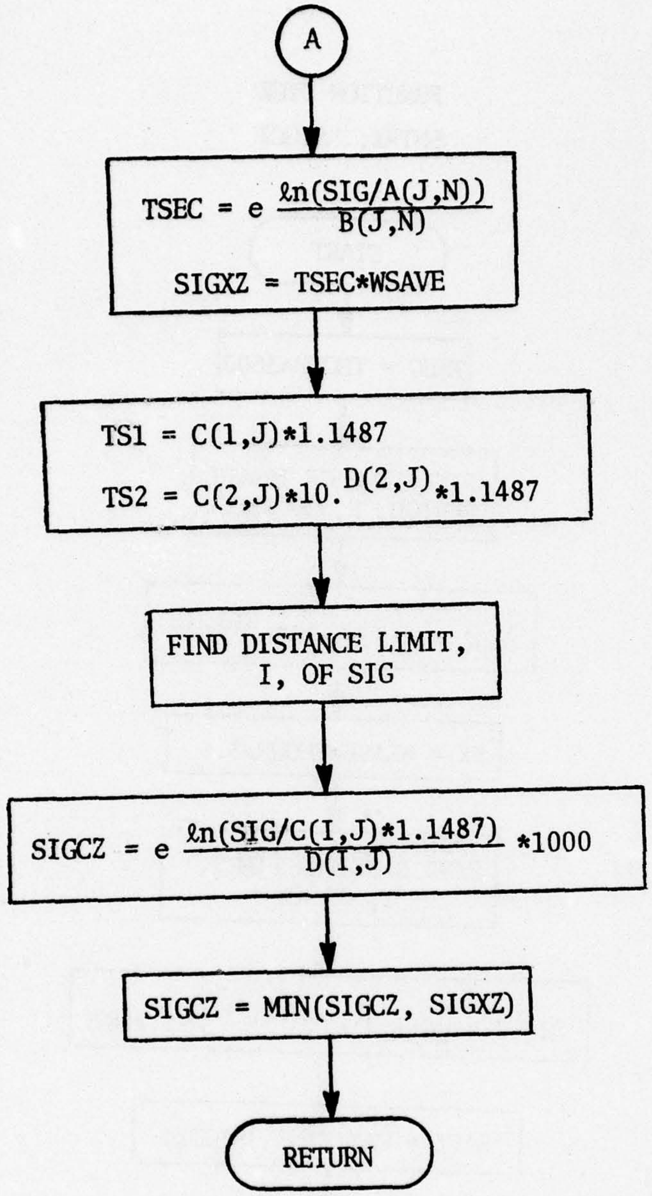
Output:

1. SIGZ = vertical dispersion in meters
2. SIGCZ = virtual distance in meters

FUNCTION SIGZ

ENTRY: SIGCZ




A
TSEC = e $\frac{\ln(SIG/A(J,N))}{B(J,N)}$
SIGXZ = TSEC*WSAVE
TS1 = C(1,J)*1.1487
TS2 = C(2,J)*10.$^{D(2,J)}$*1.1487
FIND DISTANCE LIMIT, I, OF SIG
SIGCZ = e $\frac{\ln(SIG/C(I,J)*1.1487)}{D(I,J)}$ *1000
SIGCZ = MIN(SIGCZ, SIGXZ)
RETURN

```
      FUNCTION SIGZ(J,THOUR)
C
C   THIS FUNCTION COMPUTES THE VERTICAL DISPERSION COEFFICIENT
C   IN METERS
C
      COMMON /WDUN/ WSAVE
      DIMENSION C(3,6),D(3,6),A(6,6),B(6,6),CK(6,6)
      DIMENSION TIME(6)
      DATA TIME/   300.,1000.,3000.,10000.,30000.,172000./
      DATA A/.17122,.27668,.41219,.51921,.50963,.47639,
     1       .11062,.39953,.41219,.57145,.76485,.71936,
     2       .01338,.16640,.41219,1.0813,1.9467,2.3901,
     3       .01338,.16640,.41219,2.2830,2.9850,3.8684,
     3       .01338,.16640,.41219,2.3333,5.7990,16.897,
     3       .01338,.16640,.41219,5.6801,14.599,64.577/
      DATA B/1.2098,1.0572,.92365,.84130,.79689,.76308,
     1       1.2864,.99275,.92365,.82449,.72571,.69082,
     2       1.5922,1.1195,.92365,.73217,.59047,.51700,
     3       1.5922,1.1195,.92365,.63883,.53708,.45686,
     4       1.5922,1.1195,.92365,.63646,.46497,.29621,
     5       1.5922,1.1195,.92365,.55016,.37541,.16667/
      DATA C/470.,470.,470.,110.,110.,110.,60.,60.,60.,33.,33.,40.,
     . 21.5,21.5,36.,14.,14.,23.5/
      DATA D/1.67,2.13,2.13,1.,1.09,1.09,0.92,0.92,0.92,0.80,0.61,0.53,
     . 0.70,0.56,0.35,0.78,0.53,0.30/
      DATA CK/
     1 170.,      115.,      80.,       63.,      48.,      37.,
     2 800.,      380.,      243.25,    170.,     115.,     85.,
     3 4600.,     1300.,     671.,      380.,     220.,     150.,
     4 31279.,    5002.,     2040.32,   820.,     420.,     260.,
     5 179855.2,17111.38,5628.47, 1650.,    700.,     358.,
     6 2900444.,120872.5,28241.86,4312.55,1348.32,481.58/
C
      TSEC=THOUR*3600.
      DO 10 N=1,6
      IF(TSEC.LE.TIME(N))GO TO 20
   10 CONTINUE
      N=6
C
C   TIME OF TRAVEL SHOULD BE LESS THAN 172000 SEC. OR APPROX. 2 DAYS
C
   20 CONTINUE
      SIGZ=(A(J,N)*TSEC**B(J,N))
      XX=WSAVE*THOUR*3.6
      I=1
      IF(XX.GT.1.) I=2
      IF(XX.GT.10.) I=3
C
C   CONVERTS FROM A 10 TO 20 MIN. SAMPLING TIME
C   1.1487 = 2**.2, THE 1/5 POWER LAW ONLY APPLIES UP TO 20 MIN.
C   SAMPLING TIMES
C
      SIGTZ=(C(I,J )*XX**D(I,J ))*1.1487
      SIGZ=AMAX1(SIGZ,SIGTZ)
      RETURN
      ENTRY SIGCZ(J,SIG)
C
C   AT THIS ENTRY THE DISTANCE OR TRAVEL TIME CORESPONDING TO THE
C   INPUT VALUE OF THE VERTICAL DISPERSION IS CALCULATED AND
C   RETURNED AS DISTANCE IN METERS
C
      DO 110 N=1,6
```

```
      IF (SIG.LE.CK(J,N)) GO TO 120
  110 CONTINUE
      N=6
  120 CONTINUE
      TSEC=EXP(ALOG(SIG/A(J,N))/B(J,N))
      SIGXZ=TSEC**SAVE
      TS1=C(1,J)*1.1487
      TS2=C(2,J)*10.**D(2,J)*1.1487
      I=3
      IF (SIG.LT.TS2) I=2
      IF (SIG.LT.TS1) I=1
      SIGCZ=EXP(ALOG(SIG/(C(I,J)*1.1487))/D(I,J))*1000.
      SIGCZ=AMIN1(SIGCZ,SIGXZ)
      RETURN
      END
```

SUBROUTINE SOURCE

Purpose:

To position the master source tape to read the airbase and environ source inventory data and to call the subroutines which compute the emission rates in micrograms per second at the airbase and environ sources.

Input:

JFLAG, a parameter to indicate whether the diurnal distribution used is input, default or the same as previous hour.

Output:

A statement indicating the diurnal distribution used.

Subroutines Called:

ABPTAR, ABARAR, ABLNAR, ENARAY

```
      SUBROUTINE SOURCE
C
C     THIS ROUTINE SERVES AS A DRIVER TO CALL SUBROUTINES
C     WHICH COMPUTE THE EMISSION RATES IN MICROGRAMS
C     PER SECOND AT THE AIRBASE AND ENVIRON SOURCES
C
      COMMON / DEFALT / ITAPE
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      DIMENSION NAME(2)
      DATA NAME /4H1/12,4H1     /
      IF (IONCE.EQ.0) GO TO 30
      IEND=1
      IST=1
      GO TO 40
   30 IEND=0
      IONCE=1
      IST=0
   40 CONTINUE
      DAYS=NODAYS
      IF (IST.EQ.1) GO TO 3
    1 FORMAT (I4)
      READ 1, JFLAG
      IF(JFLAG) 8,7,3
    7 PRINT 5
    5 FORMAT(32H0INPUT DIURNAL DISTRIBUTION USED)
      GO TO 4
    8 I=1
      IF (NODAYS.EQ.365) I=2
      PRINT 9,NAME(I)
    9 FORMAT(34H0DEFAULT DIURNAL DISTRIBUTION USED/5X,12HHOUR = 1/24,5X,
     .10HDAY = 1/7,5X,8HMONTH = A4,1H,5X,12HUNIFRC = 0.1)
      GO TO 4
    3 PRINT 6
    6 FORMAT(39H0DIURNAL DISTRIBUTION SAME AS LAST HOUR)
      GO TO 10
    4 IF(IEND.EQ.0) GO TO 12
   11 READ (ITAPE,END=12)
      GO TO 11
   12 IEND=1
      CALL ABPTAR
      CALL ABARAR
      CALL ABLNAR
      CALL ENARAY
   10 REWIND ITAPE
      RETURN
      END
```

SUBROUTINE STPOL1

Purpose:

To calculate pollutant concentrations from point and area sources.

Input:

1. Location and conditions at point or area source
2. Location of receptors
3. Meteorological conditions

Output:

Concentration of pollutants at each receptor

Procedure:

1. For area sources determine average diameter, effective stack height, and initial values of horizontal and vertical dispersion. Also consider effects of downwash on these.
2. For point sources determine plume rise by calling PLRISE.
3. Consider effects of wind speed at height of source.
4. Calculate crosswind and downwind components for the source.
5. Calculate time required for plume to travel from virtual point source to actual location of true source by PSEUDO routine.
6. For each receptor, calculate crosswind and downwind components.
7. Consider the relative location of receptor with respect to source and, if necessary, calculate coupling coefficient using TRAN routine.
8. For each pollutant, add in concentration determined.

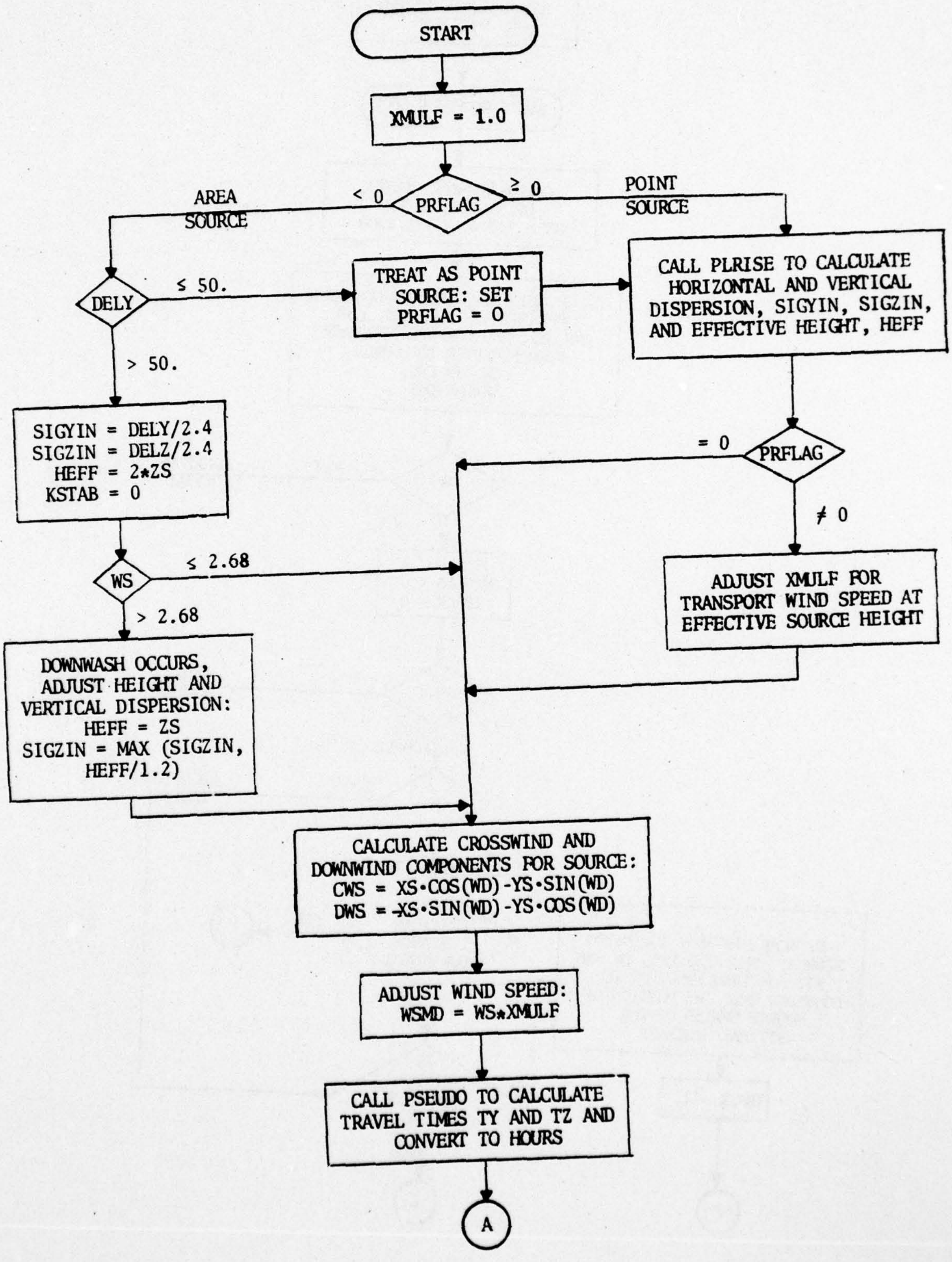
Subroutines Called:

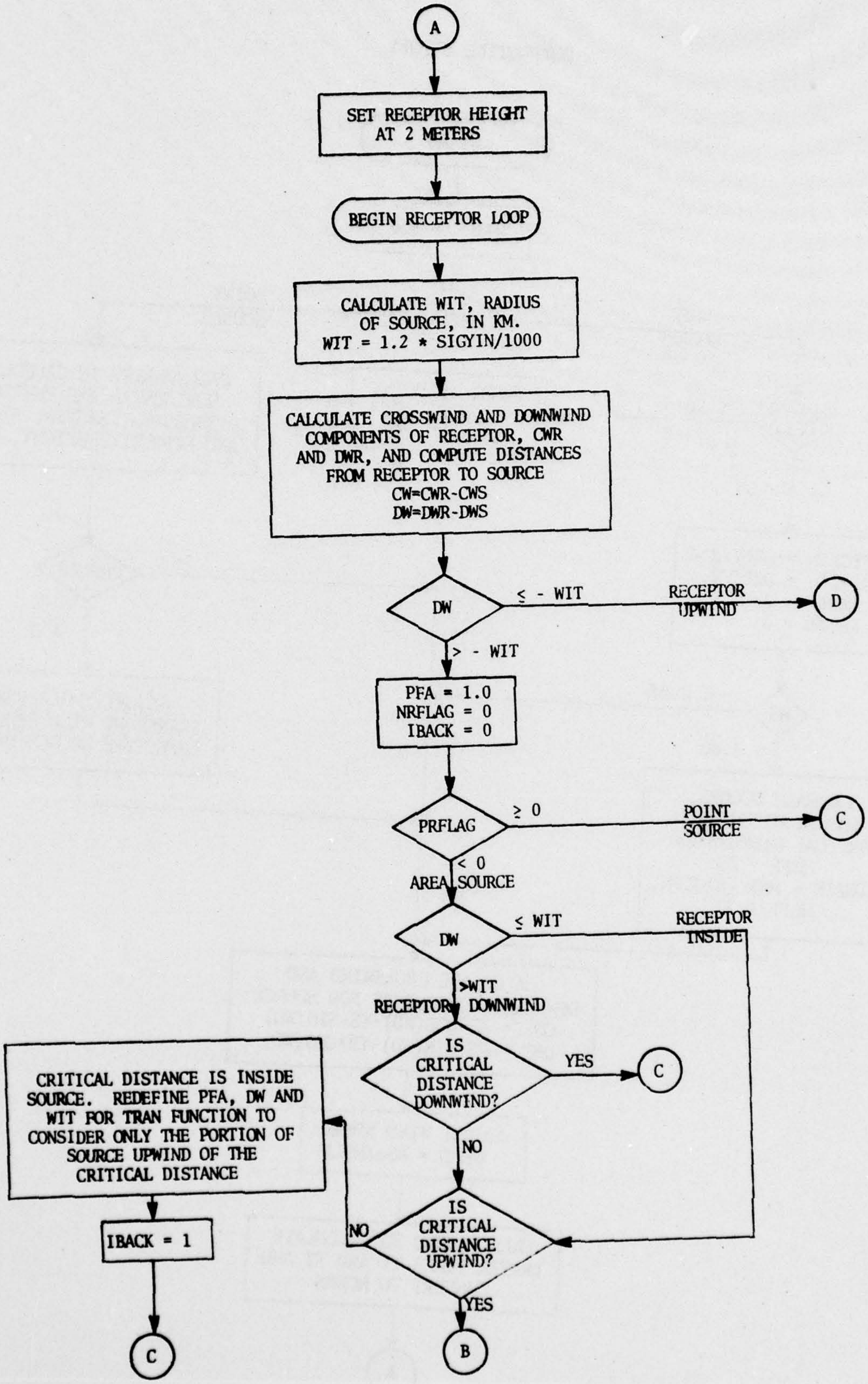
PLRISE, PSEUDO

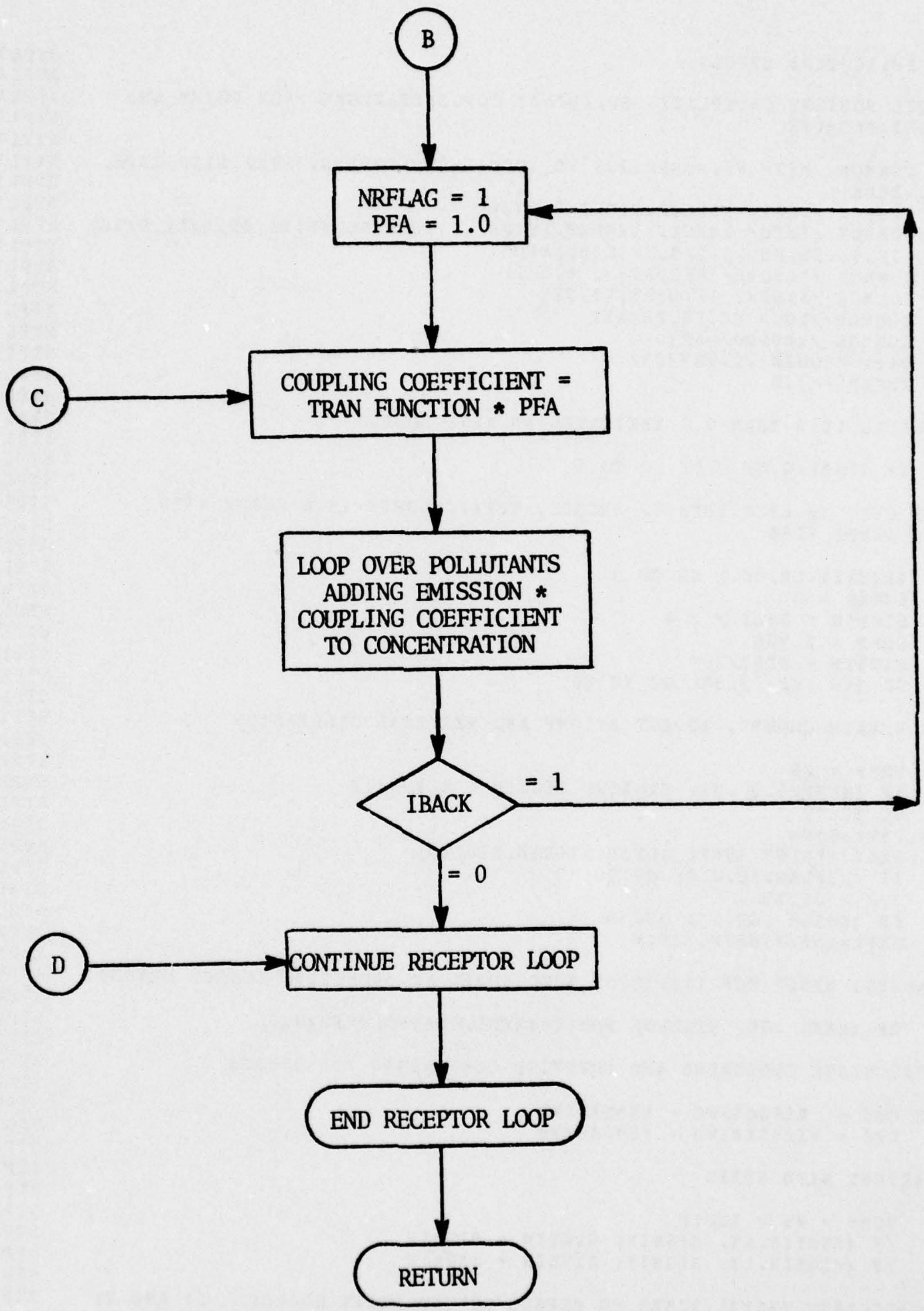
Function Called:

TRAN

SUBROUTINE STPOL1




A
SET RECEPTOR HEIGHT AT 2 METERS
BEGIN RECEPTOR LOOP
CALCULATE WIT, RADIUS OF SOURCE, IN KM.
WIT = 1.2 * SIGYIN/1000
CALCULATE CROSSWIND AND DOWNWIND COMPONENTS OF RECEPTOR, CWR AND DWR, AND COMPUTE DISTANCES FROM RECEPTOR TO SOURCE
CW=CWR-CWS
DW=DWR-DWS
DW
≤ - WIT
RECEPTOR UPWIND
D
> - WIT
PFA = 1.0
NRFLAG = 0
IBACK = 0
PRFLAG
≥ 0
POINT SOURCE
C
< 0
AREA SOURCE
DW
≤ WIT
RECEPTOR INSIDE
>WIT
RECEPTOR DOWNWIND
IS CRITICAL DISTANCE DOWNWIND?
YES
C
NO
CRITICAL DISTANCE IS INSIDE SOURCE. REDEFINE PFA, DW AND WIT FOR TRAN FUNCTION TO CONSIDER ONLY THE PORTION OF SOURCE UPWIND OF THE CRITICAL DISTANCE
IBACK = 1
C
NO
IS CRITICAL DISTANCE UPWIND?
YES
B



B
NRFLAG = 1
PFA = 1.0
C
COUPLING COEFFICIENT = TRAN FUNCTION * PFA
LOOP OVER POLLUTANTS ADDING EMISSION * COUPLING COEFFICIENT TO CONCENTRATION
IBACK
= 1
= 0
D
CONTINUE RECEPTOR LOOP
END RECEPTOR LOOP
RETURN

```
      SUBROUTINE STPOL1
C
C   THIS ROUTINE CALCULATES POLLUTANT CONCENTRATIONS FROM POINT AND
C   AREA SOURCES
C
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE     ,HTAERO,XS,YS,ZS,DELY,DELZ,
     . TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /XTRAN/ XL,WSMD,TY,TZ
      COMMON /LOC/ DW,CW,ZR,WIT
      COMMON /WNDPRO/ XP(6)
      DATA SIGMIN /2.083333/
      XMULF = 1.0
C
C   PRFLAG LESS THAN 0.0 INDICATES AN AREA SOURCE
C
      IF (PRFLAG.GE.0.0) GO TO 5
C
C   IF DELY IS LESS THAN 50 METERS, TREAT SOURCE AS A POINT WITH
C   NO PLUME RISE
C
      IF(DELY.LE.50.) GO TO 4
      KSTAB = 0
      SIGYIN = DELY / 2.4
      HEFF = 2.*ZS
      SIGZIN = DELZ/2.4
      IF (WS .LE. 2.68) GO TO 10
C
C   DOWNWASH OCCURS, ADJUST HEIGHT AND VERTICAL DISPERSION
C
      HEFF = ZS
      IF (HEFF/1.2 .GT. SIGZIN) SIGZIN = HEFF/1.2
      GO TO 10
    4 PRFLAG=0.
    5 CALL PLRISE (HEFF,KSTAB,SIGZIN,SIGYIN)
      IF (PRFLAG.EQ.0.0) GO TO 10
      JJJ = JSTAB
      IF (KSTAB .GE. 1) JJJ=5
      HEFL=AMIN1(HEFF,305.)
C
C   ADJUST XMULF FOR TRANSPORT WIND SPEED AT EFFECTIVE SOURCE HEIGHT
C
      IF (HEFL .GT. HTAERO) XMULF=(HEFL/HTAERO)**XP(JJJ)
C
C   CALCULATE CROSSWIND AND DOWNWIND COMPONENTS FOR SOURCE
C
   10 CWS =  XS*COSEWD - YS*SINEWD
      DWS = -XS*SINEWD - YS*COSEWD
C
C   ADJUST WIND SPEED
C
      WSMD = WS * XMULF
      IF (SIGYIN.LT. SIGMIN) SIGYIN = SIGMIN
      IF (SIGZIN.LT. SIGMIN) SIGZIN = SIGMIN
C
C   CALCULATE TRAVEL TIMES TO PSEUDO UPWIND POINT SOURCE.  TY AND TZ
C   ARE RETURNED AS DISTANCES IN METERS AND CONVERTED TO EQUIVALENT
C   TIMES IN HOURS
C
```

```
      CALL PSEUDO(SIGYIN,WSMD,SIGZIN,TY,TZ)
      TY=TY/WSMD/3600.
      TZ=TZ/WSMD/3600.
      ZR = 2.
C
C   BEGIN RECEPTOR LOOP
C
      DO 20 NR=1,NRECEP
C
C   WIT IS RADIUS OF SOURCE IN KILOMETERS
C
      WIT = 1.2 * SIGYIN / 1000.
C
C   CALCULATE CROSSWIND AND DOWNWIND COMPONENTS OF RECEPTOR AND
C   COMPUTE DISTANCES IN KILOMETERS FROM RECEPTOR TO SOURCE
C
      CWR = RECEP(1,NR)*COSEWD - RECEP(2,NR)*SINEWD
      CW = CWR - CWS
      DWR = -RECEP(1,NR) * SINEWD - RECEP(2,NR) * COSEWD
      DW = DWR - DWS
C
C   PFA IS FRACTION OF TOTAL AREA SOURCE BEING TREATED
C
      IF (DW.LE.-WIT) GO TO 20
C
C   IS RECEPTOR UPWIND OF SOURCE?
C
      PFA = 1.0
      NRFLAG=0
      IBACK=0
      IF (PRFLAG.GE.0.0) GO TO 16
C
C   IS RECEPTOR INSIDE SOURCE?
C
      IF(DW.LE.WIT) GO TO 15
C
C   IS CRITICAL DISTANCE DOWNWIND OF DOWNWIND EDGE OF SOURCE?
C
      IF((DWR-XL).GE.(DWS+WIT)) GO TO 16
C
C   IS CRITICAL DISTANCE UPWIND OF UPWIND EDGE OF SOURCE?
C
   15 IF((DWR-XL).LT.(DWS-WIT)) GO TO 17
C
C   CRITICAL DISTANCE IS INSIDE SOURCE.  REDEFINE PFA, DW AND WIT
C   FOR TRAN FUNCTION TO CONSIDER ONLY THE PORTION OF SOURCE
C   UPWIND OF THE CRITICAL DISTANCE
C
      PFA=(DW-XL+WIT)/(2.*WIT)
      DWSS=((DWS-WIT)+(DWR-XL))/2.
      DW=DWR-DWSS
      WIT=DW-XL
      IBACK=1
      GO TO 16
   17 NRFLAG=1
      PFA=1.
C
C   CALL TRAN FUNCTION TO DETERMINE COUPLING COEFFICIENTS
C
   16 CUPCOE = TRAN (KSTAB,HEFF,NRFLAG,IBACK ) * PFA
C
C   ADD EMISSIONS TIMES COUPLING COEFFICIENT TO CONCENTRATIONS
```

```
C   AT ALL RECEPTORS
C
      DO 18 IPOL = 1,NPOL
   18 RECDAT(ITYPE,IPOL,NR) = RECDAT(ITYPE,IPOL,NR) + EMIS(IPOL) *CUPCOE
      IF(IBACK.EQ.1) GO TO 17
   20 CONTINUE
      RETURN
      END
```

SUBROUTINE STPOL2

Purpose:

To prepare the data required by the line source model and, for each receptor, to call the model and then add the pollutant concentrations calculated to the accumulated totals at that receptor.

Input:

1. Source parameters for the current line.
2. Wind speed and direction, and lid height.

Output:

Accumulated pollutant concentrations at all receptors.

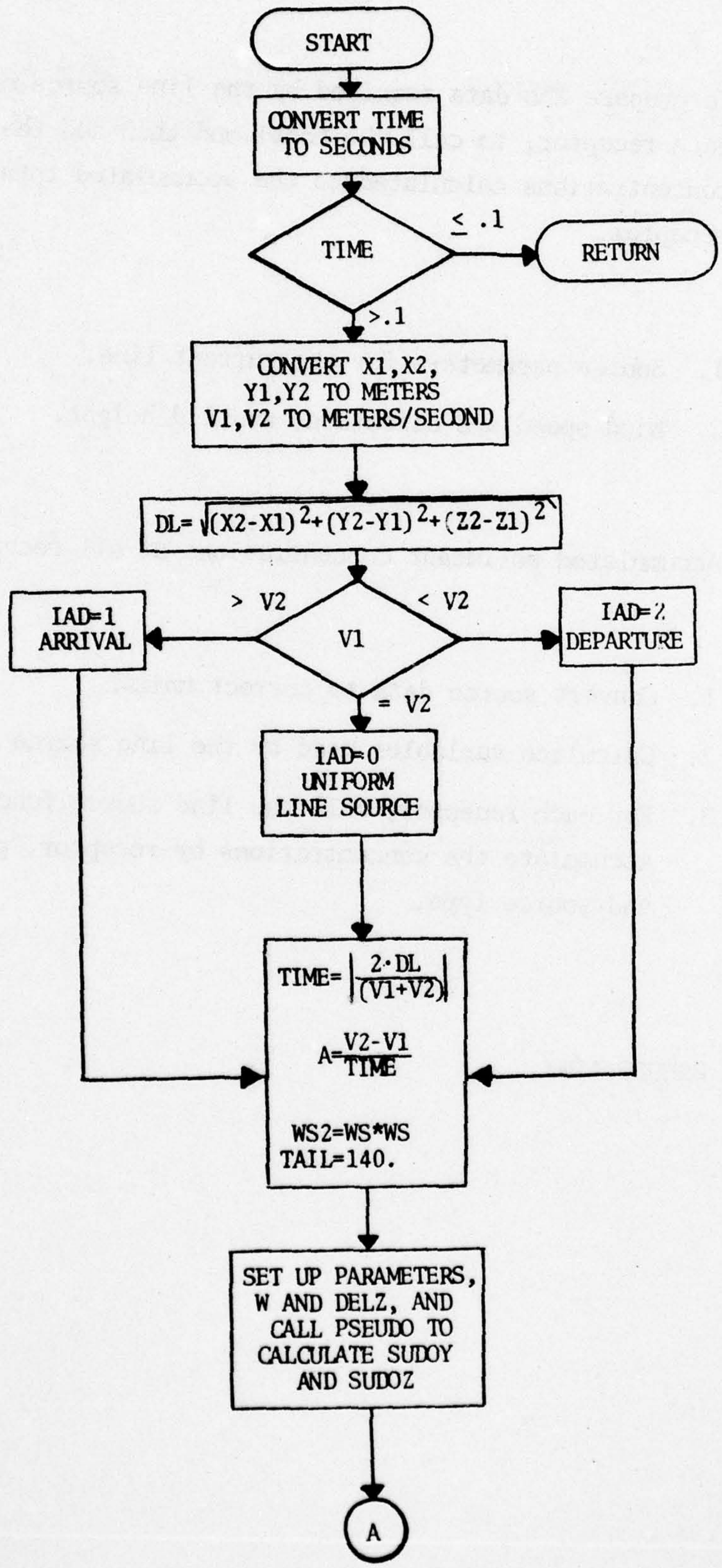
Procedure:

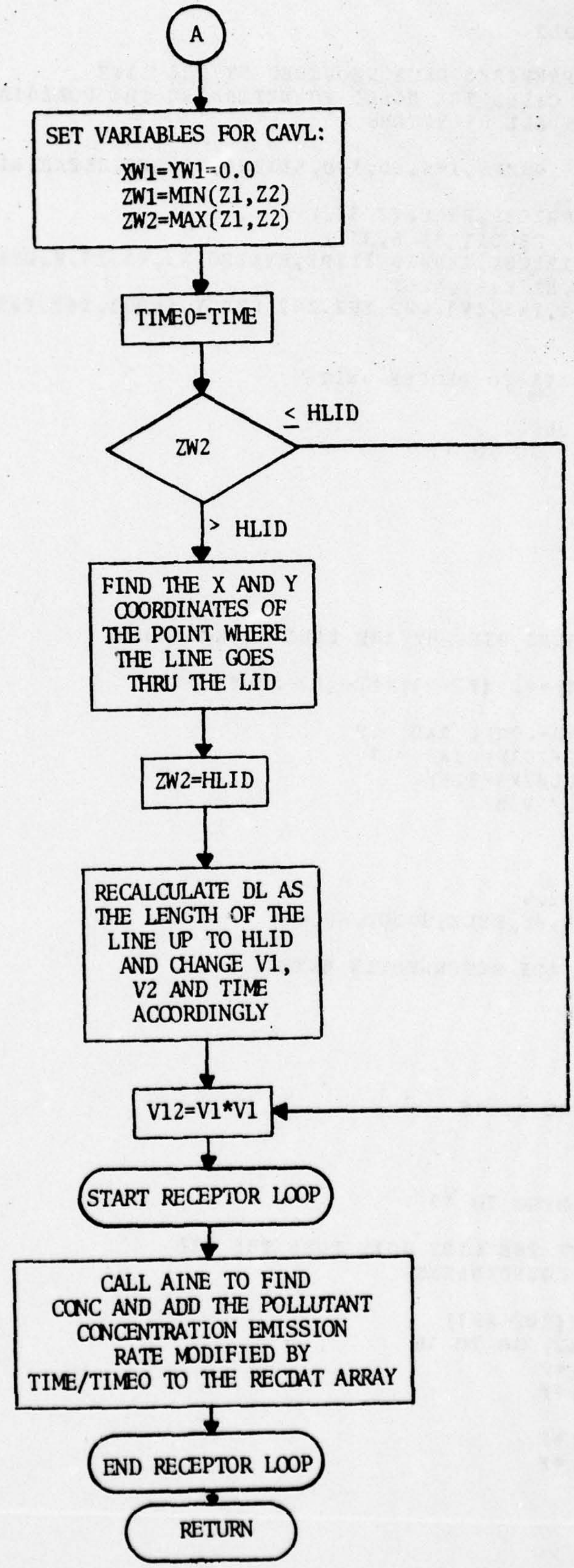
1. Convert source data to correct units.
2. Calculate variables used by the line source model.
3. For each receptor, call the line source function and accumulate the concentrations by receptor, pollutant, and source type.

Subroutines Called:

PSEUDO,AINE

SUBROUTINE STPOL2




A
SET VARIABLES FOR CAVL:
XW1=YW1=0.0
ZW1=MIN(Z1,Z2)
ZW2=MAX(Z1,Z2)
TIME0=TIME
ZW2
≤ HLID
> HLID
FIND THE X AND Y COORDINATES OF THE POINT WHERE THE LINE GOES THRU THE LID
ZW2=HLID
RECALCULATE DL AS THE LENGTH OF THE LINE UP TO HLID AND CHANGE V1, V2 AND TIME ACCORDINGLY
V12=V1*V1
START RECEPTOR LOOP
CALL AINE TO FIND CONC AND ADD THE POLLUTANT CONCENTRATION EMISSION RATE MODIFIED BY TIME/TIME0 TO THE RECDAT ARRAY
END RECEPTOR LOOP
RETURN

AD-A046 348 ARGONNE NATIONAL LAB ILL F/G 13/2

AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)

APR 77 D J BINGAMAN

UNCLASSIFIED CEEDO-TR-76-34 NL

3 OF 3

AD
A046348

END
DATE
FILMED
12-77
DDC

```
      SUBROUTINE STPOL2
C
C    THIS SUBROUTINE PREPARES DATA REQUIRED BY THE LINE
C    SOURCE MODEL AND CALLS THE MODEL TO DETERMINE THE POLLUTANT
C    CONCENTRATIONS AT ALL RECEPTORS
C
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,
     . TEMK
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,
     . V1,V2,DL,TIME,EMIS(6),NPOL
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,
     . WS2,WSC,RR,SP
C
C    CONVERT SOURCE DATA TO PROPER UNITS
C
      TIME = TIME * 3600.
      IF(TIME.LE.0.1) GO TO 11
      X1 = X1 * 1000.
      Y1 = Y1 * 1000.
      X2 = X2 * 1000.
      Y2 = Y2 * 1000.
      V1 = V1 / 3.6
      V2 = V2 / 3.6
C
C    CALCULATE VARIABLES USED BY THE LINE SOURCE MODEL
C
      DL=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)
      IAD = 0
      IF (V1 .LT. (V2-.01)) IAD = 2
      IF (V1 .GT.(V2+.01)) IAD = 1
      TIME = ABS(2*DL/(V1+V2))
      A = (V2 - V1) / TIME
      WS2 = WS * WS
      TAIL = 140.
      W = W / 2.4
      DELZ = DELZ / 2.4
      CALL PSEUDO (W,WS,DELZ,SUDOY,SUDOZ)
C
C    SUDOY AND SUDOZ ARE RETURNED IN METERS
C
      XW1=0.
      YW1=0.
      ZW1=Z1
      ZW2=Z2
      IF(Z1.LE.Z2) GO TO 15
      ZW1=Z2
      ZW2=Z1
   15 TIMEO=TIME
      IF(ZW2.LE.HLID)GO TO 18
C
C    FIND POINT WHERE THE LINE GOES THRU THE LID
C    AND CHANGE THE COORDINATES
C
      F=(HLID-ZW1)/(ZW2-ZW1)
      IF (Z1 .GT. Z2) GO TO 16
      X2=X1+(X2-X1)*F
      Y2=Y1+(Y2-Y1)*F
      GC TC 17
   16 X1=X2+(X1-X2)*F
      Y1=Y2+(Y1-Y2)*F
```

```
 17   ZW2 = HLID
C
C   RECALCULATE THE LENGTH OF THE LINE UP TO HLID AND
C   CHANGE VELOCITIES ACCORDINGLY
C
      DLSQ = (X1-X2)**2 + (Y1-Y2)**2 + (ZW1-ZW2)**2
      DL = SQRT(DLSQ)
      IF (Z2 .GT. Z1) V2 = SQRT(V1*V1+2.*A*DL)
      IF (Z2 .LT. Z1) V1 = SQRT(V2*V2-2.*A*DL)
      TIME = 2* DL / (V1+V2)
 18   V12 = V1 * V1
C
C   CALL THE LINE FUNCTION TO DETERMINE POLLUTANT CONCENTRATIONS
C   AT ALL RECEPTORS
C
      DO 10 IRECEP=1,NRECEP
      CONC = AINE(WD)
      DO 10 IPOL=1,NPOL
   10 RECDAT (ITYPE,IPOL,IRECEP) = RECDAT(ITYPE,IPOL,IRECEP) +
     - EMIS(IPOL) * CONC * TIME / TIMEO
   11 CONTINUE
      RETURN
      END
```

FUNCTION TRAN

Purpose:

To calculate the coupling coefficient at a receptor due to a point or area source.

Input:

1. Meteorological conditions: wind speed, stability, mixing height, critical distance for mixing.
2. Source parameters: initial horizontal and vertical dispersion; effective stack height; pseudo transport times corresponding to the dispersions; plume height flag, KSTAB; area source flags, NRFLAG and IBACK.

Output:

Point or area source coupling coefficient, TRAN.

Procedure:

1. If the effective stack height exceeds the mixing height, then the stability index is reassigned according to the KSTAB flag, the lid is set at 3050 meters and the critical distance at 100 meters.
2. For sources with NRFLAG=0, compute the travel time for z dispersion from the center and that for y dispersion from the downwind edge of the source. Then the effects of ground and sky lid are treated by the image method, with up to 6 terms included in the coupling coefficient.
3. For area sources with NRFLAG=1, the travel times from the upwind and downwind edges of the source are determined on the basis of receptor location relative to the source. These plus the pseudo travel time, TZ, due to the z spread are used to compute the z-dispersion coefficients σ_z (T1) and σ_z (T2). The y-dispersion coefficient σ_y (TT) is determined on the basis of the pseudo travel time, TY, due to the y-spread plus the travel time from the downwind edge

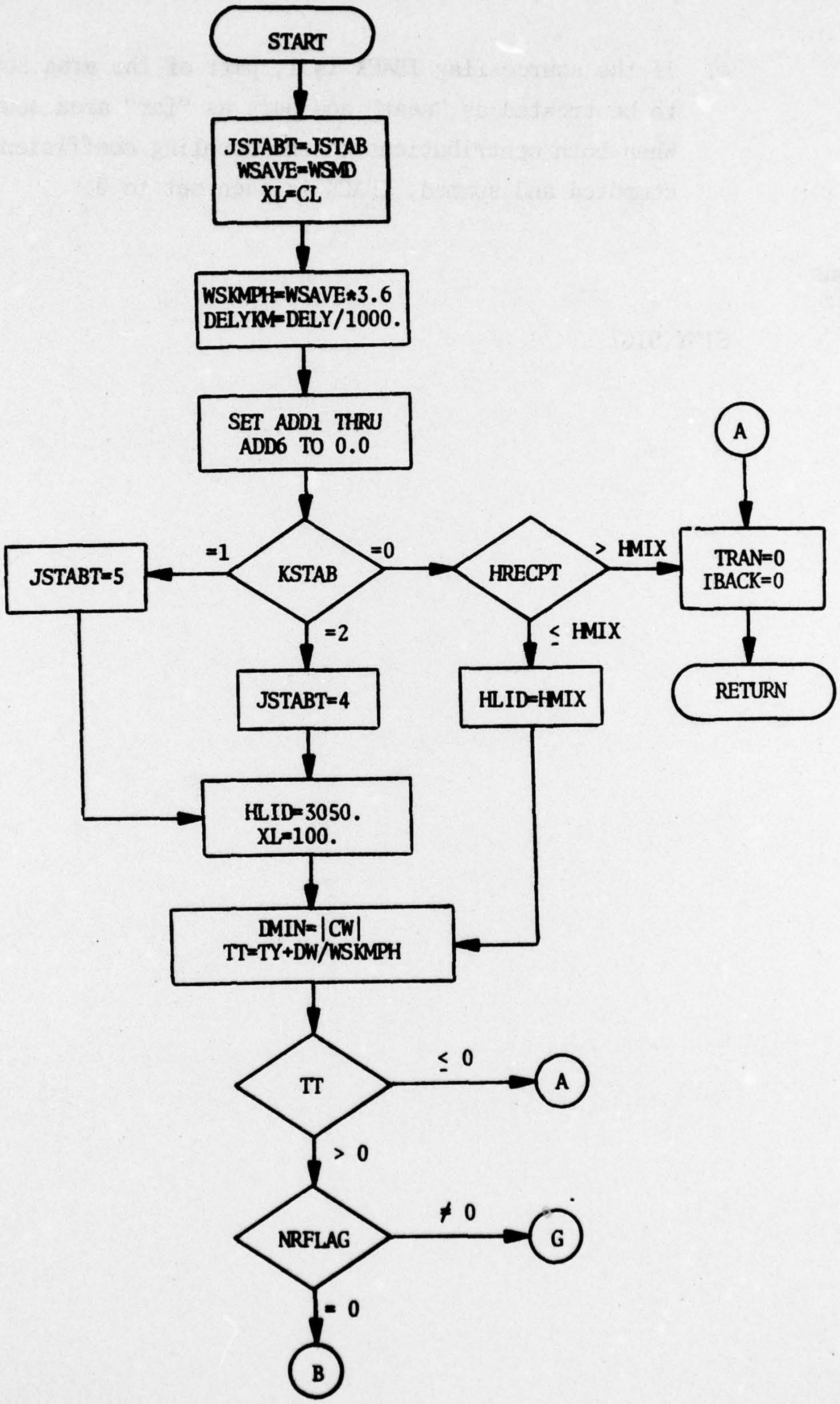
to the receptor. Then the coupling coefficient is computed using the integrated expression for "near" source.

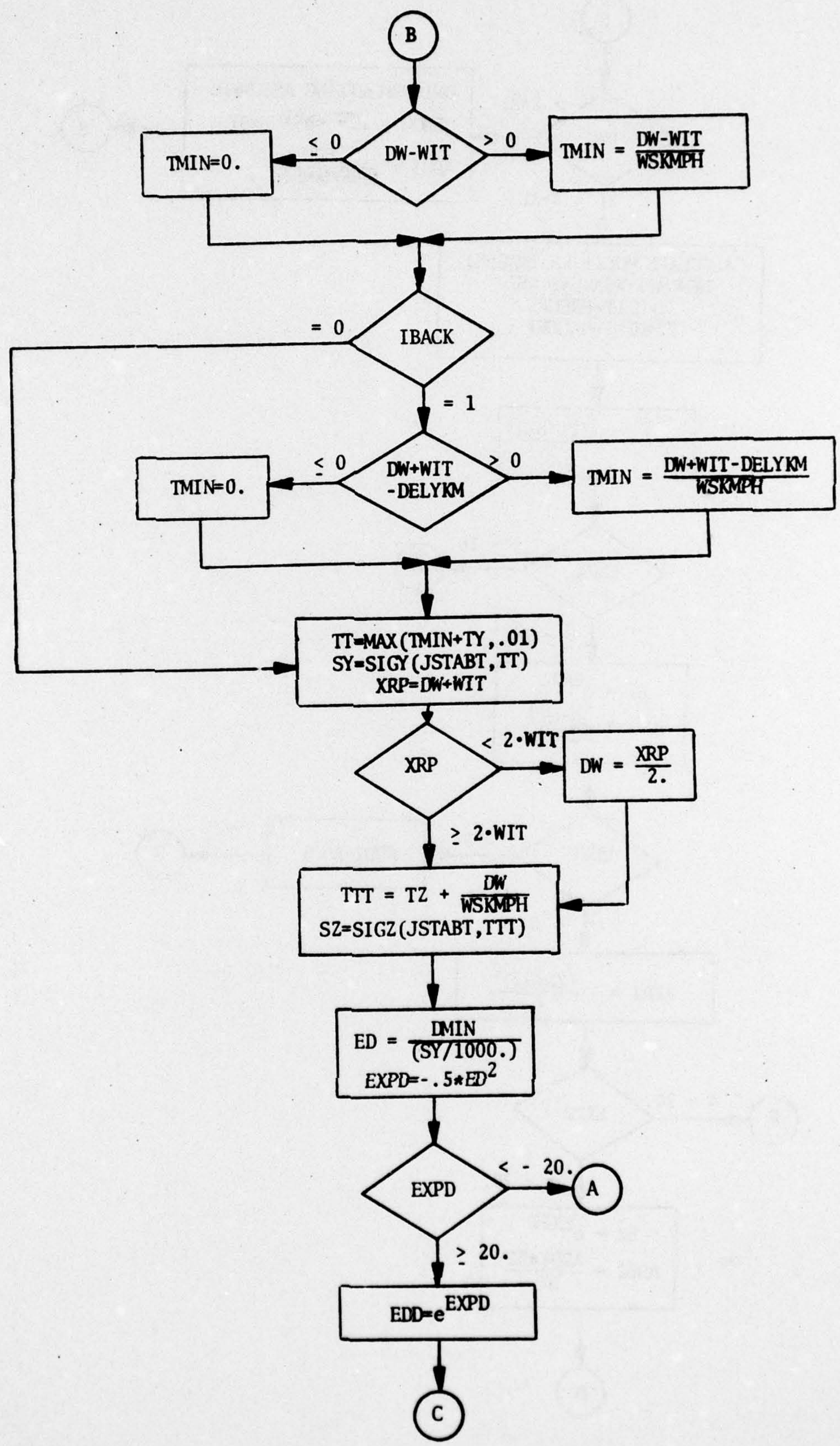
4. If the source flag IBACK is 1, part of the area source is to be treated as "near" and part as "far" area sources. When both contributions to the coupling coefficient are computed and summed, IBACK is then set to 0.

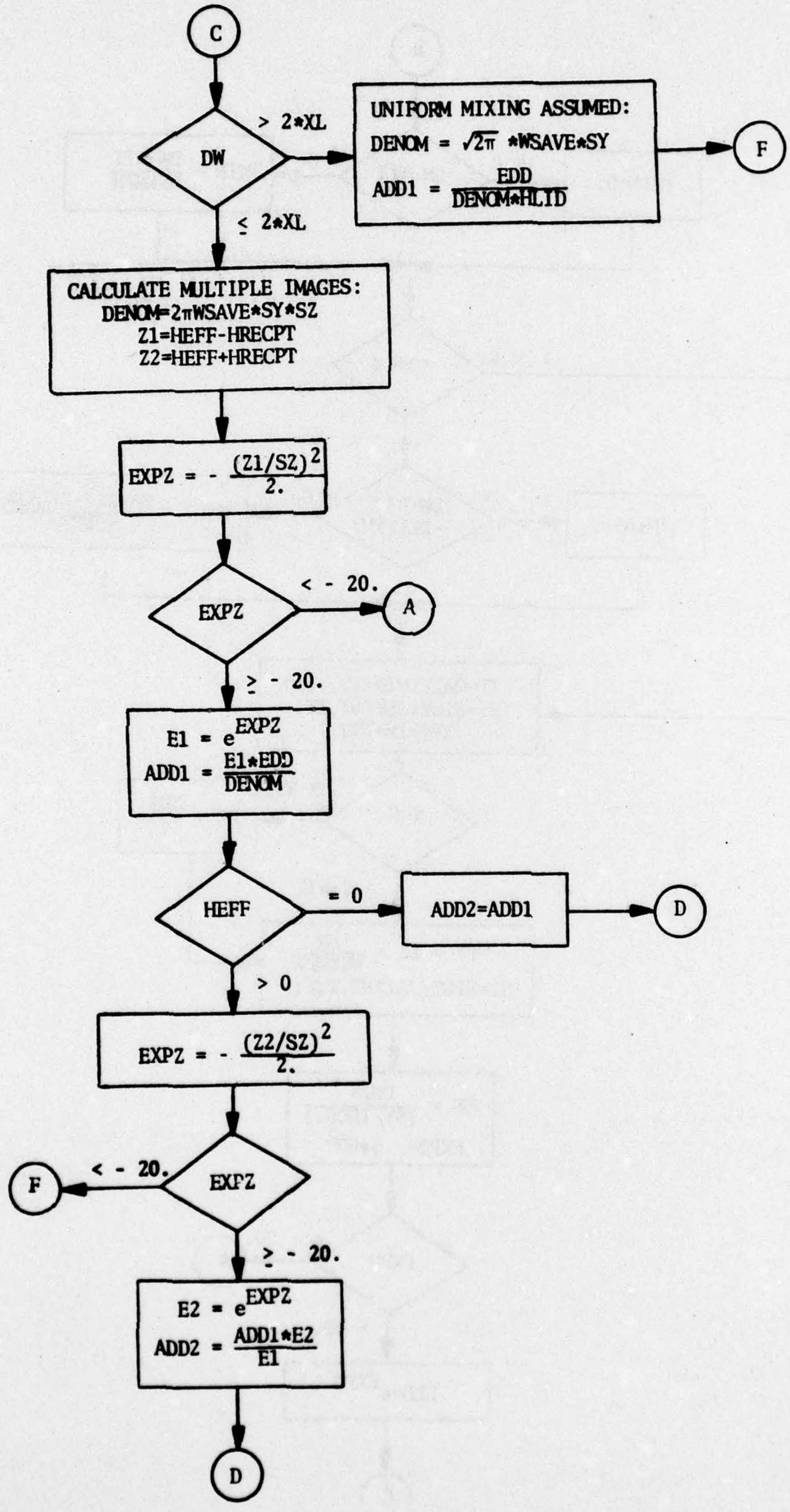
Functions
Called:

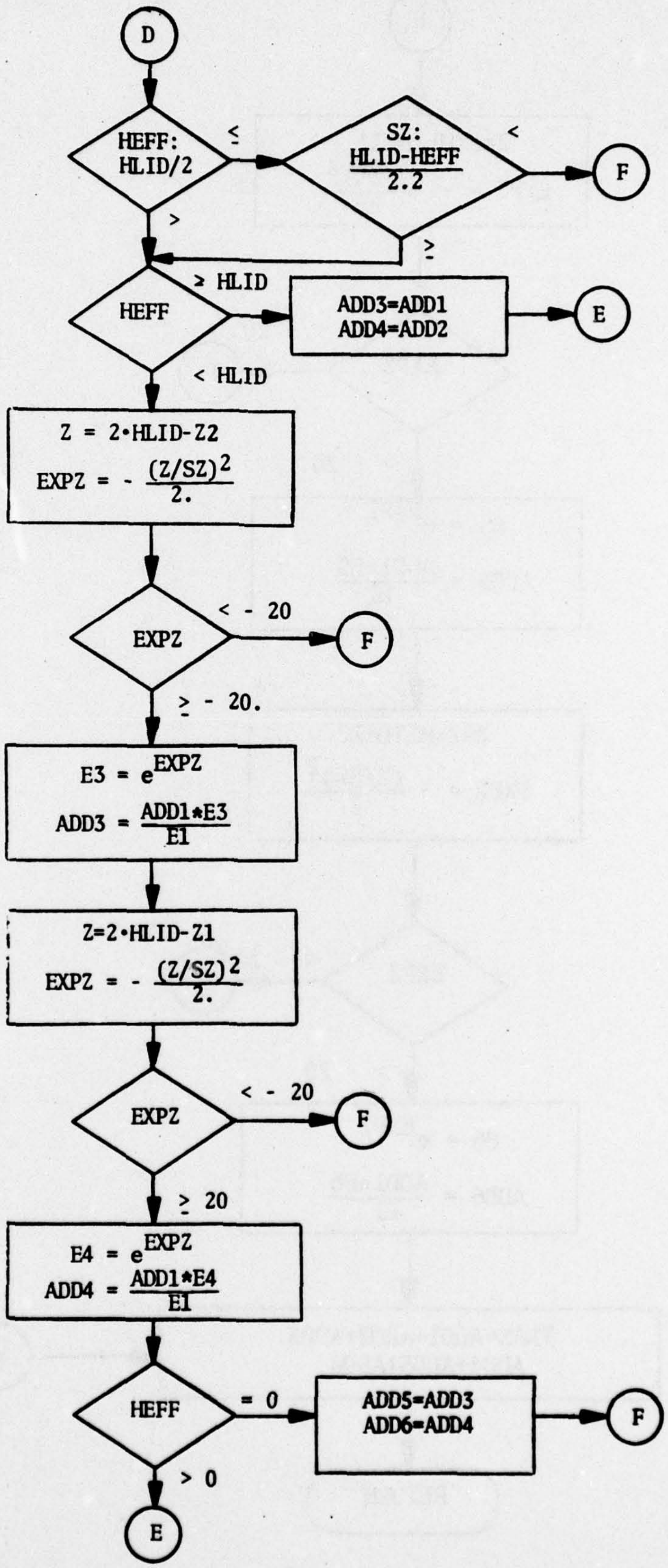
SIGY,SIGZ

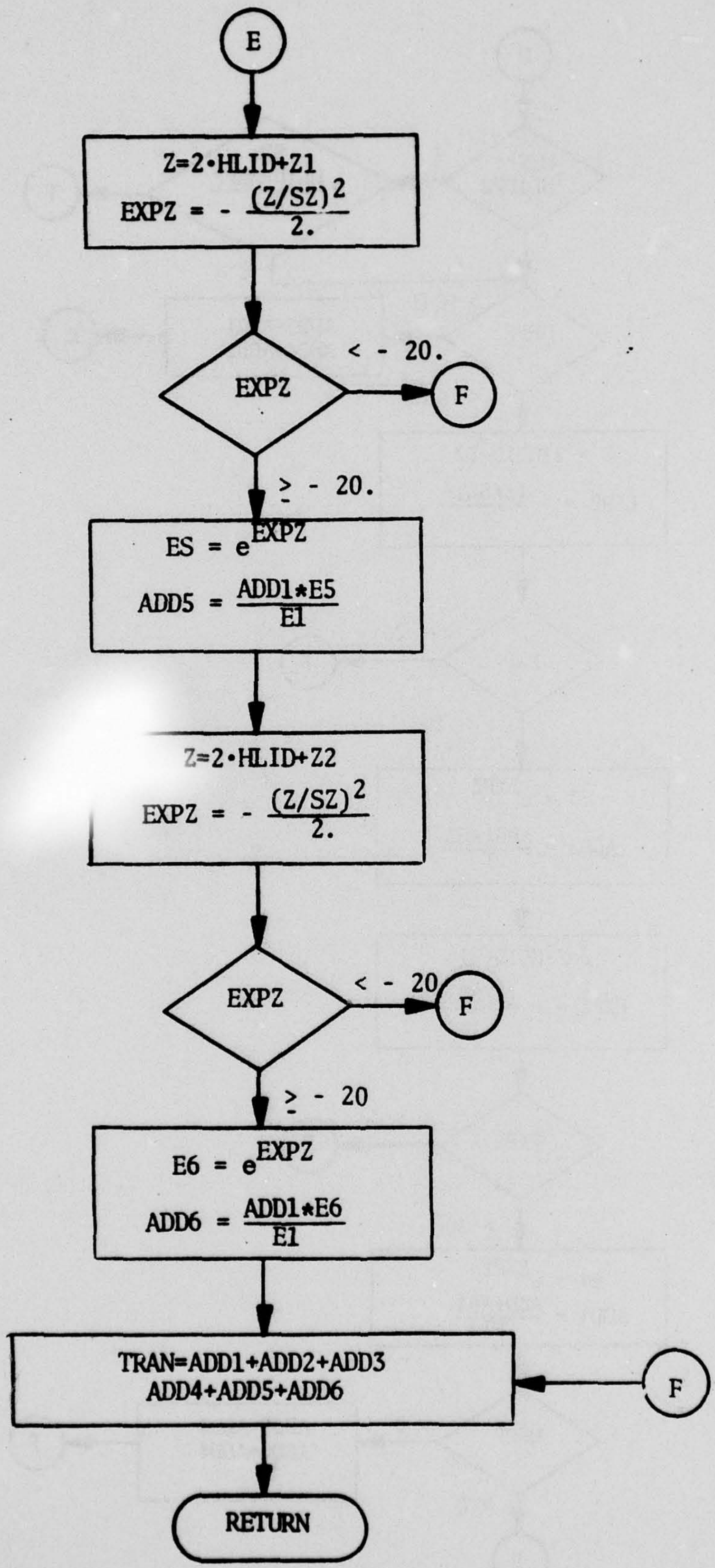
FUNCTION TRAN (KSTAB, HEFF, NRFLAG, IBACK)

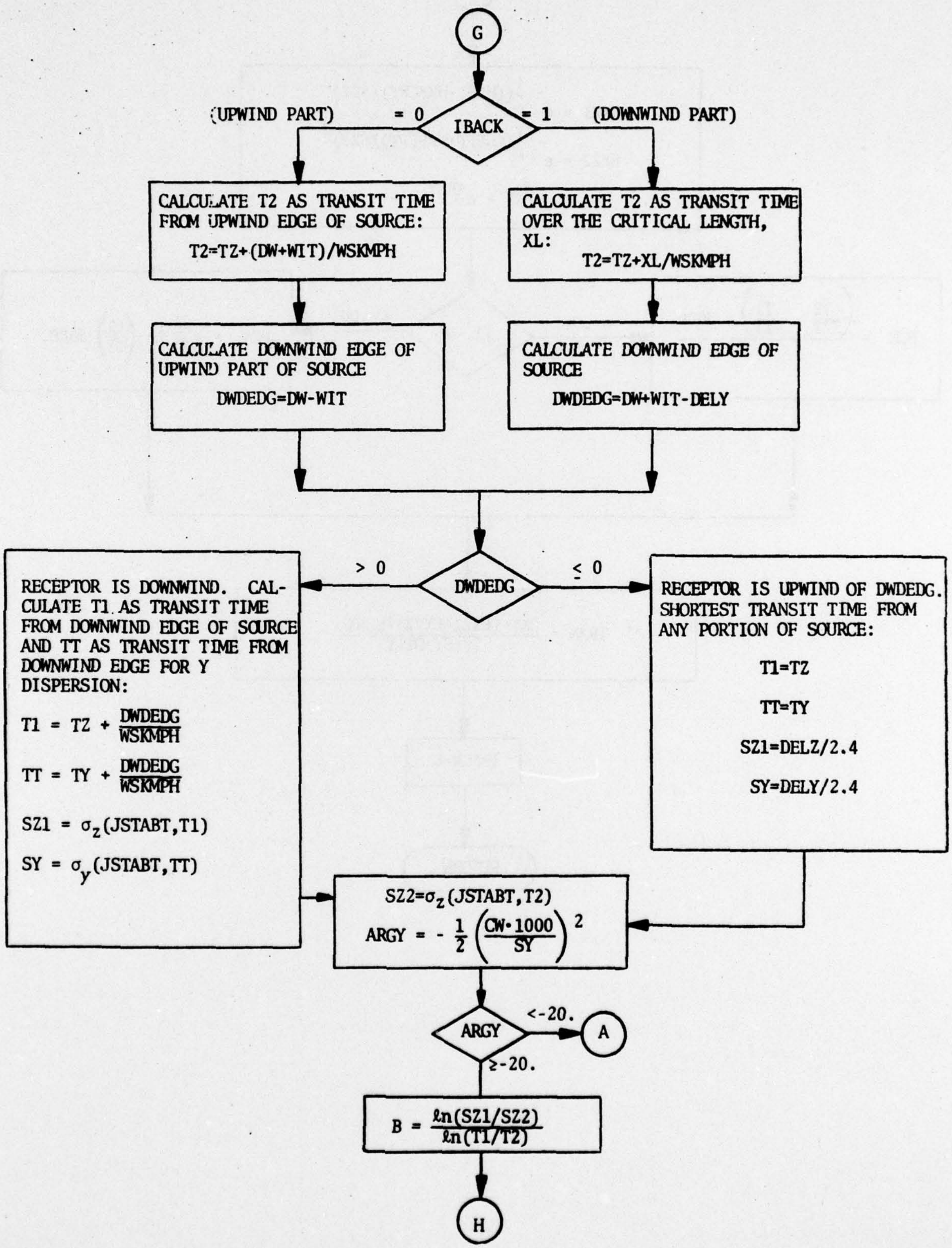


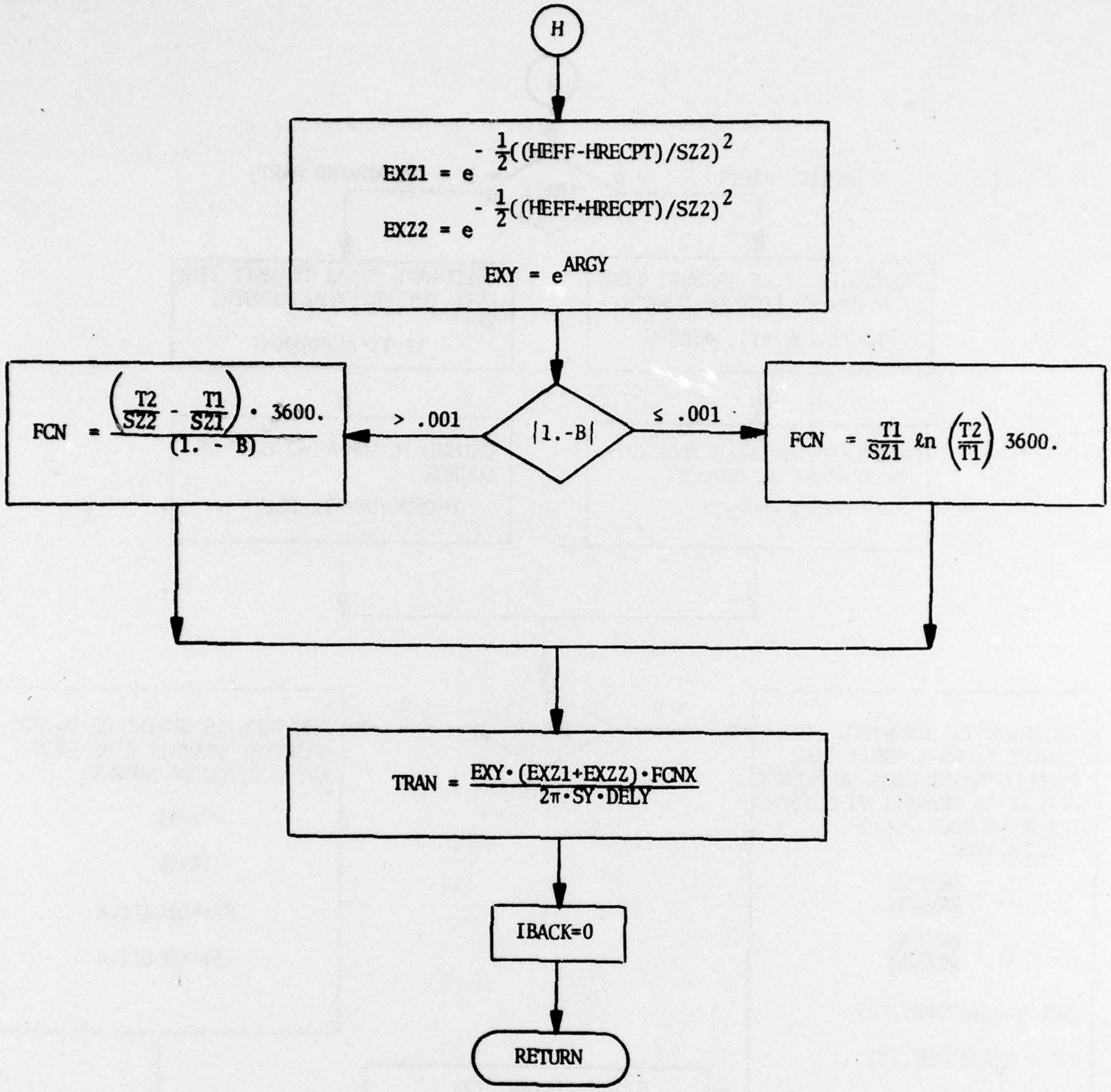

B
DW-WIT
≤ 0
TMIN=0.
> 0
TMIN = DW-WIT / WSKMPH
IBACK
= 0
= 1
DW+WIT -DELYKM
≤ 0
TMIN=0.
> 0
TMIN = DW+WIT-DELYKM / WSKMPH
TT=MAX(TMIN+TY,.01)
SY=SIGY(JSTABT,TT)
XRP=DW+WIT
XRP
< 2·WIT
DW = XRP / 2.
≥ 2·WIT
TTT = TZ + DW / WSKMPH
SZ=SIGZ(JSTABT,TTT)
ED = DMIN / (SY/1000.)
EXPD=-.5*ED²
EXPD
< - 20.
A
≥ 20.
EDD=e^EXPD
C


C
DW
> 2*XL
≤ 2*XL
UNIFORM MIXING ASSUMED:
DENOM = √2π *WSAVE*SY
ADD1 = EDD / DENOM*HLID
F
CALCULATE MULTIPLE IMAGES:
DENOM=2πWSAVE*SY*SZ
Z1=HEFF-HRECPT
Z2=HEFF+HRECPT
EXPZ = - (Z1/SZ)^2 / 2.
EXPZ
< - 20.
A
≥ - 20.
E1 = e^EXPZ
ADD1 = E1*EDD / DENOM
HEFF
= 0
ADD2=ADD1
D
> 0
EXPZ = - (Z2/SZ)^2 / 2.
EXPZ
< - 20.
F
≥ - 20.
E2 = e^EXPZ
ADD2 = ADD1*E2 / E1
D


D
HEFF: HLID/2
≤
>
SZ: (HLID-HEFF)/2.2
<
≥
F
HEFF
≥ HLID
< HLID
ADD3=ADD1
ADD4=ADD2
E
Z = 2•HLID-Z2
EXPZ = - (Z/SZ)²/2.
EXPZ
< - 20
≥ - 20.
F
E3 = e^EXPZ
ADD3 = ADD1*E3/E1
Z=2•HLID-Z1
EXPZ = - (Z/SZ)²/2.
EXPZ
< - 20
≥ 20
F
E4 = e^EXPZ
ADD4 = ADD1*E4/E1
HEFF
= 0
> 0
ADD5=ADD3
ADD6=ADD4
F
E


E
Z=2•HLID+Z1
EXPZ = - (Z/SZ)^2 / 2.
EXPZ
< - 20.
F
≥ - 20.
ES = e^EXPZ
ADD5 = ADD1*E5 / E1
Z=2•HLID+Z2
EXPZ = - (Z/SZ)^2 / 2.
EXPZ
< - 20
F
≥ - 20
E6 = e^EXPZ
ADD6 = ADD1*E6 / E1
TRAN=ADD1+ADD2+ADD3
ADD4+ADD5+ADD6
F
RETURN


G
(UPWIND PART)
= 0
IBACK
= 1
(DOWNWIND PART)
CALCULATE T2 AS TRANSIT TIME FROM UPWIND EDGE OF SOURCE:
T2=TZ+(DW+WIT)/WSKMPH
CALCULATE T2 AS TRANSIT TIME OVER THE CRITICAL LENGTH, XL:
T2=TZ+XL/WSKMPH
CALCULATE DOWNWIND EDGE OF UPWIND PART OF SOURCE
DWDEDG=DW-WIT
CALCULATE DOWNWIND EDGE OF SOURCE
DWDEDG=DW+WIT-DELY
> 0
DWDEDG
≤ 0
RECEPTOR IS DOWNWIND. CALCULATE T1 AS TRANSIT TIME FROM DOWNWIND EDGE OF SOURCE AND TT AS TRANSIT TIME FROM DOWNWIND EDGE FOR Y DISPERSION:
$T1 = TZ + \frac{DWDEDG}{WSKMPH}$
$TT = TY + \frac{DWDEDG}{WSKMPH}$
$SZ1 = \sigma_z(JSTABT,T1)$
$SY = \sigma_y(JSTABT,TT)$
RECEPTOR IS UPWIND OF DWDEDG. SHORTEST TRANSIT TIME FROM ANY PORTION OF SOURCE:
T1=TZ
TT=TY
SZ1=DELZ/2.4
SY=DELY/2.4
$SZ2=\sigma_z(JSTABT,T2)$
$ARGY = -\frac{1}{2}\left(\frac{CW\cdot 1000}{SY}\right)^2$
ARGY
<-20.
A
≥-20.
$B = \frac{\ln(SZ1/SZ2)}{\ln(T1/T2)}$
H


H
EXZ1 = e^{-½((HEFF-HRECPT)/SZ2)²}
EXZ2 = e^{-½((HEFF+HRECPT)/SZ2)²}
EXY = e^{ARGY}
|1.-B|
> .001
≤ .001
FCN = (T2/SZ2 - T1/SZ1) · 3600. / (1. - B)
FCN = T1/SZ1 ℓn (T2/T1) 3600.
TRAN = EXY·(EXZ1+EXZ2)·FCNX / 2π·SY·DELY
IBACK=0
RETURN

```
      FUNCTION TRAN (KSTAB,HEFF,NRFLAG,IBACK)
C
C     THIS FUNCTION CALCULATES THE COUPLING COEFFICIENT
C     AT A RECEPTOR DUE TO A POINT OR AREA SOURCE
C
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE     ,HTAERO,XS,YS,ZS,DELY,DELZ,
     . TS,VS,DS,HB,PRFLAG,EMIS( 8),NPOL
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HMIX,TEMP,
     . T3MF
      COMMON /XTRAN/ CL,WSMD,TY,TZ
      COMMON/WDUH/WSAVE
      COMMON/LOC/DW,CW,HRECPT,WIT
      DATA SQ2PI /2.5066283/
      XL = CL
      JSTABT=JSTAB
      WSAVE=WSMD
      WSKMPH=WSAVE*3.6
      DELYKM=DELY/1000.
      ADD1=0.
      ADD2=0.
      ADD3=0.
      ADD4=0.
      ADD5=0.
      ADD6=0.
C
      IF (KSTAB.GT.0) GO TO 121
C
C     KSTAB=0, PLUME IS BELOW THE LID, IF RECEPTOR IS ABOVE
C     LID, TRAN = 0.
C
      IF (HRECPT.GT.HMIX) GO TO 76
      HLID=HMIX
      GOTO 140
C
C     ASSUME ARBITRARILY HIGH LID HEIGHT FOR TWO CASES WHEN:
C       KSTAB = 1, PLUME IS INITIALLLY ABOVE THE LID
C       KSTAB = 2, PLUME WILL PENETRATE THE LID
C     ASSIGN STABILITY CLASSES 5 AND 4 RESPECTIVELY
C
  121 JSTABT=6-KSTAB
      HLID=3050.
      XL = 100.
  140 CONTINUE
      DMIN=ABS(CW)
      TT=TY+DW/WSKMPH
      IF (TT.LE.0.) GO TO 76
      IF (NRFLAG.NE.0) GO TO 143
C
C     NRFLAG=0, EFFECTS OF GROUND AND SKY LID ARE TREATED
C     BY THE MULTIPLE IMAGE METHOD, WITH UP TO 6 TERMS
C     INCLUDED IN THE COUPLING COEFFICIENT
C
      TMIN = 0.
      IF (DW-WIT.GT.0.) TMIN=(DW-WIT)/WSKMPH
      IF (IBACK.EQ.0) GO TO 131
C
C     IBACK=1, RECEPTOR IS WITHIN CRITICAL DOWNWIND DISTANCE.
C     TREAT PORTIONS OF SOURCE UPWIND OF CRITICAL LENGTH
C     FROM RECEPTOR
C
      IF (DW+WIT-DELYKM) 132,132,133
  132 TMIN=0.
```

```
      GO TO 131
  133 THIN=(DW+WIT-DELYKM)/WSKMPH
  131 TT=AMAX1(THIN+TY,.01)
      SY=SIGY(JSTABT,TT)
      XRP = WIT + DW
      IF (XRP.LT.2.*WIT) DW=XRP/2.
      TTT=TZ+DW/WSKMPH
      SZ=SIGZ(JSTABT,TTT)
      ED=DMIN/(SY/1000.)
      EXPD=-.5*ED*ED
      IF (EXPD.LT.-20.) GO TO 76
      EDD=EXP(EXPD)
      IF (DW.GT.2.*XL) GO TO 153
C
C   CALCULATE MULTIPLE IMAGES
C
      DENOM=6.2831853*WSAVE*SY*SZ
C
C    IMAGES 1 AND 2:  GROUND REFLECTION OF SOURCE
C
      Z2=HEFF+HRECPT
      Z1=HEFF-HRECPT
      EXPZ=-(Z1/SZ)**2/2.
      IF (EXPZ.LT.-20.) GO TO 76
      E1=EXP(EXPZ)
      ADD1=E1*EDD/DENOM
      IF (HEFF.GT.0.0) GO TO 171
      ADD2=ADD1
      GOTO 172
  171 CONTINUE
      EXPZ=-(Z2/SZ)**2/2.
      IF (EXPZ.LT.-20.) GO TO 61
      E2=EXP(EXPZ)
      ADD2=ADD1*E2/E1
C
C    IMAGES 3 AND 4:  REFLECTION ABOUT HLID OF SOURCE
C
  172 CONTINUE
      IF (HEFF.LE.HLID/2.0.AND.SZ.LT.(HLID-HEFF)/2.2) GO TO 61
      IF (HEFF.LT.HLID) GO TO 174
      ADD3=ADD1
      ADD4=ADD2
      GOTO 173
  174 CONTINUE
      Z=2.*HLID-Z2
      EXPZ=-(Z/SZ)**2/2.
      IF (EXPZ.LT.-20.) GO TO 61
      E3=EXP(EXPZ)
      ADD3=ADD1*E3/E1
      Z=2.*HLID-Z1
      EXPZ=-(Z/SZ)**2/2.
      IF (EXPZ.LT.-20.) GO TO 61
      E4=EXP(EXPZ)
      ADD4=ADD1*E4/E1
      IF (HEFF.GT.0.0) GO TO 173
      ADD5=ADD3
      ADD6=ADD4
      GO TO 61
C
C    IMAGES 5 AND 6:  REFLECTION ABOUT HLID OF FIRST BELOW GROUND IMAGE
C
  173 CONTINUE
```

```
      Z=2.*HLID+Z1
      EXPZ=-(Z/SZ)**2/2.
      IF (EXPZ.LT.-20.) GO TO 61
      E5=EXP(EXPZ)
      ADD5=ADD1*E5/E1
      Z=2.*HLID+Z2
      EXPZ=-(Z/SZ)**2/2.
      IF (EXPZ.LT.-20.) GO TO 61
      E6=EXP(EXPZ)
      ADD6=ADD1*E6/E1
      GO TO 61
C
C   UNIFORM MIXING ASSUMED
C
  153 DENOM=SQ2PI*WSAVE*SY
      ADD1=EDD/(DENOM*HLID)
   61 CONTINUE
      TRAN=ADD1+ADD2+ADD3+ADD4+ADD5+ADD6
      RETURN
C
   76 TRAN=0.
      IBACK=0
      RETURN
C
C   NRFLAG=1, RECEPTOR IS CLOSE TO SOURCE
C
  143 IF (IBACK.EQ.1) GO TO 144
C
C   UPWIND PART OF SOURCE
C
      T2=(DW+WIT)/WSKMPH+TZ
      DWDEDG=DW-WIT
      GO TO 145
C
C   DOWNWIND PART OF SOURCE
C
  144 T2=TZ+XL/WSKMPH
      DWDEDG=DW+WIT-DELYKM
  145 IF (DWDEDG.LE.0.) GO TO 146
C
C   RECEPTOR IS DOWNWIND .  FIND TRANSIT TIMES FROM
C   DOWNWIND EDGE OF SOURCE
C
      T1=TZ+DWDEDG/WSKMPH
      TT=TY+DWDEDG/WSKMPH
      SZ1=SIGZ(JSTABT,T1)
      SY=SIGY(JSTABT,TT)
      GO TO 147
C
C   RECEPTOR IS UPWIND.  FIND SHORTEST TRANSIT TIME FROM
C   ANY PORTION OF SOURCE
C
  146 T1=TZ
      TT=TY
      SZ1=DELZ/2.4
      SY=DELY/2.4
C
C   COMPUTE COUPLING COEFFICIENT USING INTEGRATED
C   EXPRESSION FOR 'NEAR' SOURCE
C
  147 SZ2=SIGZ(JSTABT,T2)
  155 ARGY=-(CW*1000./SY)**2/2.
```

```
      IF (ARGY.LT.-20.) GO TO 76
      B=ALOG(SZ1/SZ2)/ALOG(T1/T2)
      EXZ1=EXP(-((HEFF-HRECPT)/SZ2)**2/2.)
      EXZ2=EXP(-((HEFF+HRECPT)/SZ2)**2/2.)
      EXY=EXP(ARGY)
      IF (ABS(1.-B).LE..001) GO TO 2
      FCNX=(T2/SZ2-T1/SZ1)*3600./(1.-B)
      GO TO 3
C
C    SPECIAL CASE  FOR B = 1.
C
    2 FCNX=T1/SZ1*ALOG(T2/T1)*3600.
C
    3 TRAN=EXY*(EXZ1+EXZ2)*FCNX/(6.2831853*SY*DELY)
      IBACK=0
      RETURN
      END
```

REFERENCES

1. Bingaman, D. J. and L. E. Wangen, "Air Quality Assessment Model for Air Force Operations - Source Emissions Inventory Computer Code Documentation." Air Force Civil and Environmental Engineering Development Office report number, CEEDO-TR-76-33.

2. Menicucci, D. F., "Air Quality Assessment Model (AQAM) Data Reduction and Operations Guide," Air Force Weapons Laboratory report number AFWL-75-307, Oct 76.

3. Rote, Donald M., and L. E. Wangen, "A Generalized Air Quality Assessment Model for Air Force Operations - Technical Report," Air Force Weapons Laboratory report number AFWL-TR-74-304, February 1975.

INITIAL DISTRIBUTION

USAF/PREVP	3
TAC/DEEV	2
CINCUSAFE/Surgeon	1
USAFA/DEV	1
AFIT/DEM	1
OEHL/CC	1
OEHL/OL-AA	5
ADTC/CSV	1
DDC/TCA	12
USA Environ Hygn Agency	1
USA CERL	1
WESTDIV (Code 09BE)	1
Technology Transfer/EPA	1
R&D/EPA	1
NARF/Code 64270	2
Nav Post Grad Sch	2
EESD/DR	5
US Env Prt Agency	2
FAA/AEQ	1
Det 1 HQ ADTC/WE	2
Det 1 HQ ADTC/EC	1
Det 1 HQ ADTC/ECA	7
AFCEC/DEE	3
Dr Delaney	1
AUL/AUL-LSE-70-239	1
Det 1 HQ ADTC/PRL	2
AFATL/DLODL	1
AFATL/DLODR	1

(The reverse of this page is blank)